ADAPTIVE ROUTING ALGORITHM FOR PRIORITY FLOWS IN A NETWORK

THESIS

Timothy J. Carbino, Captain, USAF

AFIT/GE/ENG/12-08

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/12-08

ADAPTIVE ROUTING ALGORITHM FOR PRIORITY FLOWS IN A NETWORK

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Timothy J. Carbino, BS

Captain, USAF

June 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/GE/ENG/12-08

ADAPTIVE ROUTING ALGORITHM FOR PRIORITY FLOWS IN A NETWORK

Timothy J. Carbino, BS
Captain, USAF

Approved:

Kenneth M. Hopkinson
Kenneth M. Hopkinson, PhD (Chairman) — 29 May 2012 Date

Jeffrey M. Hemmes, Lt Col, USAF(Member) — 29 May 2012 Date

Mark D. Silvius
Mark D. Silvius, Maj, USAF(Member) — 29 May 2012 Date

AFIT/GE/ENG/12-08

Abstract

This research presents the development of an Adaptive Routing Algorithm for Priority (ARAP) flows in a Network. Many devices used in today's battle space require information to function properly. The additional bandwidth requirements for such devices place an increased burden on the already congested networks in the battle space. Some devices require real time information (high priority) and other devices will not require real time information (low priority). The most popular existing protocols treat the network like an opaque entity and have little knowledge of user requirements. User requirement information is available in tactical networks and we can take advantage of the known requirements to better optimize network behavior. One such optimization is during times of congestion ARAP will enable better quality of service for higher priority information. Mechanisms such as the Network Tasking Order (NTO) and Network Weatherman (NWM), both previously developed at AFIT, can provide this information to facilitate improved network behavior. The NTO gives advance knowledge of network state allowing for improved quality of service guarantees. The NWM provides future estimates on the utilization of specific network queues.

To my wife, whose love and support has kept me going throughout my research. Without her help, I could not have made it through the long nights and exhausting days. Her help with proofreading and listening to me talk through my research has been invaluable. And to my boys, who are always my inspiration for everything I do.

Acknowledgments

I am very grateful to my academic advisor, Dr. Kenneth M. Hopkinson, for all the guidance, support, advice and kindness he has given me throughout my research. I would also like to acknowledge my committee members, Lt. Col. Jeffrey Hemmes and Maj Mark Silvius, whose suggestions helped me, refine and fine-tune my research project. I also am indebted to my research assistant, Nick Kerner, for the time he spent helping me set up simulations and writing code. His efforts saved me countless hours.

Captain Timothy J. Carbino

Table of Contents

List of Figures

List of Tables

List of Equations

List of Abbreviations

Abbreviation	Description	1st Use
ARAP	Adaptive Routing Algorithm for Priority	IV
CRS	Congressional Research Service	8
FAN	Flow Aware Network	13
GB	Gig bits	39
GIG	Global Information Grid	1
HTTP	Hypertext Transfer Protocol	5
IPv4	Internet Protocol version 4	16
IPv6	Internet Protocol version 6	16
Mbps	Megabits per second	37
MCF	Multicommodity Flow	19
MCFP	Maximum Concurrent Flow Problem	20
MPLS	Multiprotocol Label Switching	15
NS2	Network Simulator 2	40
NTO	Network Tasking Order	IV
NWM	Network Weatherman	IV
ODS	OPERATION DESERT STORM	8
OIF	OPERATION IRAQI FREEDOM	8
OSPF	Open Shortest Path First	2
QoS	Quality of Service	2
RIP	Routing Information Protocol	2

SMTP	Simple Mail Transfer Protocol	5
TcL	Tool Command Language	30
TCP	Transmission Control Protocol	5
UDP	User Datagram Protocol	5
ARAP	Adaptive Routing Algorithm for Priority	IV

ADAPTIVE ROUTING ALGORITHM FOR PRIORITY FLOWS IN A NETWORK

I. Introduction

Many of today's consumer software applications rely on real time information not stored on the host system. These applications work by sending and receiving the information they need via network connections. This is no different in the military where many of today's weapons systems also rely on real time information such as external sensors and live videos feeds. A special network called the Global Information Grid (GIG) is utilized by these systems to share information and is the military's answer to support the transition to the Network Centric Operations doctrine. The GIG is a highly complex and widespread network that enables the sharing of information between multiple users and weapons systems alike [5].

In the last two decades, the GIG has seen a dramatic increase in bandwidth demand. The majority of this increase is due to the heavy reliance of unmanned weapon systems [5]. Additionally, some defense officials feel that the increased reliance on the GIG may outpace their ability to increase the available bandwidth [5]. For example, some users can experience longer delays in sending information from source to destination, or, in some instances, information can be dropped from routers when network buffers become full, resulting in information loss.

Many of the routing policies employed today apply a shortest path philosophy that enables networks to meet many of the delay requirements for user applications. This type of system works well when the bandwidth demand is relatively low when compared to the overall bandwidth of the network. In this type of routing philosophy, some of the links and routers can experience high utilization rates, causing unnecessary delay and packet loss. At the same time, some links and routers in the system may be under-utilized. This two-pronged scenario is economically wasteful within the cyber-domain. An adaptive routing philosophy can direct some traffic on over-utilized routes and instead guide it along links that are not experiencing over-utilization. The algorithm that realizes this philosophy must be adaptive in order to capture the variability in network resource utilization.

Quality of Service

Quality of Service (QoS) in a network can be partially defined as throughput, loss rate and latency. Throughput is the amount of information that travels across a given network during a specified period. The loss rate is defined as the amount of information that does not reach its intended destination divided by the amount of information sent by the source. Latency is the time it takes the information to reach its destination once it has left the source.

The QoS that many mainstream networks provide can be considered equal opportunity because their guarantees are applied evenly to all traffic no matter the type. In fact, current Routing Information Protocol (RIP) and Open Shortest Path First (OSPF) routing protocols both use a shortest path metric to construct routing tables for a network router. In [27, 21] there is discussion that states that this can cause some problems, the first being that some links could become overused thereby causing congestion. Secondly, the capacity of the shortest path link could be met and exceeded during the same time that a longer path may be experiencing under utilization. This even distribution of QoS may not have an adverse effect on the well-being of mainstream civilian users. However, during military operations human lives depend on information carried by military networks. Delaying or dropping information has the potential to cause unforeseen harm to government interests.

By utilizing traffic engineering, a process by which one can exploit the fact that there are usually multiple paths between source-destination pairs in a network [27], network optimizations can be made with regards to QoS guarantees at the network layer with control mechanisms in place. When multiple paths exists between a source and destination, higher priority flows can be given preferential treatment on the path of their choice and low priority flows can be sent on different paths that do not adversely affect the high priority flows. This assumes that the military has the ability to categorize information into priority types that will allow military operations to benefit from the ability to distinguish between different types of priority information.

The focus of this research is to be able to give QoS guarantees to specific types of information flows in the network layer. These guarantees are in the form of delays and packet loss rate based on the type of flow. These guarantees are needed in a military environment where the timeliness and accuracy of sending and receiving different types of information can affect the outcome of the conflict.

A network that provides a diverse range of QoS to specific types of information can enable the user to ensure that time-sensitive and mission-critical information receive the resources necessary for mission success. This research proposes an adaptive routing algorithm that employs additional mechanisms to provide QoS guarantees to the higher priority information in the network. The adaptive routing algorithm is designed to operate in the Network and Link layers of the internet protocol stack. It does not utilize or build on top of any other QoS mechanisms.

Five Layers of the Internet

Basic internet architecture and its mode of operation in cyberspace is a massive undertaking to describe in its minute description; however, its general overall structure is based on the inter-relationships between five layers: Application, Transport, Network, Link and Physical. Each layer is both unique and integral in the way it supports the cyber domain. These layers work together simultaneously to help break down the complex nature of sending information from one system application to another. When these layers are combined together, they make up the five-layer Internet protocol stack, as seen in Table 1 [16].

Table 1: The Five-Layer Internet Protocol Stack [16]

Application
Transport
Network
Link
Physical

This research looks to the network layer as a place to improve upon the QoS for information flows. This is accomplished with an intelligent agent that has the ability to change the route that a flow takes based on its given priority.

A brief summary of what is to follow includes an explanation of the application and transport layers, which reside on each of the host computers. Following that is a discussion on how the link and physical layers make up the actual routers and wires that connect the routers. Lastly, a description is given of the network layer, which can be defined as the bridge between the host computer and the routers.

Application Layer

Many programs on the host computer use the application layer to communicate. Each program may use one or more application layer protocol. For instance, Microsoft Outlook utilizes the Simple Mail Transfer Protocol (SMTP), which provides the ability to transfer email messages from one computer to another or, as another example, a web browser uses Hypertext Transfer Protocol (HTTP) to interpret information from web servers to display their information on the computer screen. In order for applications installed on separate computers to communicate, they each must have a program installed on them that implements the same application layer protocol. When applications need to communicate with one another they typically need to transfer various sizes of information. When the information is too large to send in one piece, the application layer breaks the information up into smaller pieces and passes them down to the transport layer in the form of messages [16].

Transport Layer

The transport layer is responsible for relaying the application layer messages from the sending host to the receiving host. There are currently two types of transport layer protocols: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) that an application can use to share information.

TCP provides a connection-oriented service that is utilized only by the end systems and not by any of the routers or link layers that make up the switching aspects of the network. TCP also provides a guarantee of in order message delivery to applications

[16]. If information is lost in the network (i.e. dropped by a router), TCP will re-transmit the missing information until all information has reached the destination. Additionally, when TCP detects congestion on the link, a congestion control mechanism starts a back off routine, which lowers the sending rate of the information. The adverse effect of the congestion control mechanism is that it can cause a transfer of information to take longer and therefore it cannot give any QoS guarantees with respect to speed.

UDP is a connection-less service and the messages are sent on a best effort basis with no guarantees that a message will arrive at its destination. However, it does provide a constant sending rate for applications, which is used to support video or voice type flows in a network. UDP is often referred to as the best-effort protocol. The transport layer protocols then pass their information in the form of a segment down to the network layer with a destination address.

Network Layer

The basic responsibility of the network layer is to ensure that packets of information sent from a source reach its intended destination based on an address associated with the information. The network layer utilizes only one protocol called the Internet Protocol. Any internet components that have a network layer must implement this protocol [16]. The only QoS guarantee the network layer offers is for throughput and packet loss rates and these guarantees are not specific to any type of traffic.

The internet protocol is augmented by a routing protocol which determines the route packets take to reach their destinations. The routes that are calculated by the routing protocols are installed on the routers in the form of routing tables. The Network Layer, based on data provided by the routing tables, places information on the outgoing links of the router. There are several routing protocols in use today and this layer is a key focus in this research.

Link Layer

The link layer protocol is responsible for moving packets between adjacent hosts or routers within a network. These hosts and routers are also known as nodes and the terms can be used interchangeably. There are several protocols in this layer but two of the most common are Ethernet, which is used for wired connections and WiFi, which is used for wireless connections. Packets are passed back and forth between the link layer and the network layer at each node until its destination is reached.

Physical Layer

The job of the physical layer is to transport the actual information from one node to the next. As is similar in other layers, several protocols are associated with the physical layer and many of these protocols depend on the transportation medium (i.e. wireless, twisted pair or fiber optic cables).

Once the information reaches its destination, the network layers passes the information up to the transport layer where it is put back together into larger pieces prior to being passed to the application layer on the receiving computer.

Summary of Internet Layers

All these layers have specific requirements on how information is to be passed from one layer to the next. This allows the creation of new protocols as long as all protocols conform to the defined requirements.

Priority Levels

When utilizing the shortest path routing mechanism to calculate routes for flows in a network all flows are treated equally. By assigning a priority level to each packet in a flow a routing mechanism could potentially pick different routes for the flows based on priority. For example, source node *A* and a destination node *B* have two flows associated with them called *flow1* and *flow2*. *Flow1* is a high priority flow and *Flow2* is a low priority flow. In order to use priority levels, a routing mechanism that can inspect the priority level for each packet in a flow is necessary to send the two flows along different paths in the network. This can help alleviate congested links by routing lower priority flows around the congested links in the network thus giving special treatment to high priority flows. This research uses two priority levels called high and low to implement a routing protocol that gives higher priority packets better quality of service with in the network.

General Issue

The number of weapon systems in today's military requiring real-time information has skyrocketed in the past 15 years. For these systems to operate properly, they must connect to a network and receive the information that they require to operate. The addition of these advanced weapon systems have caused increased demand for network bandwidth. In fact, a Congressional Research Service (CRS) report from 2007 stated that the peak rate of information disseminated on military networks for OPERATION IRAQI FREEDOM (OIF) was approximately 30 times higher than that of OPERATION DESERT STORM (ODS) [5]. The increase in network bandwidth is primarily due to the addition of weapon systems that require real-time information. During OIF many of the networks that these devices depended on were deficient in available bandwidth and, as a result, communication officers had to disconnect network cables so that only high priority information could gain access to the network [5].

Military operations can benefit from a routing algorithm that can take into account user requirements to help alleviate network congestion where possible. By placing information into high and low categories, the network can leverage the categories to automate the stopping or slowing of low priority information during times of high network utilization. Through automation, the same officers that were left unplugging network cables during OIF to prevent lower priority information from getting on the network can now be utilized to complete other tasks.

Problem Statement

Can higher priority information experience better quality of service through implementation of an adaptive routing algorithm that utilizes network predictive mechanisms to help route information flows in the network? Lower priority flows will be allowed to continue provided there is an alternate path. If no other path can be found than the lower priority flow will be paused.

Hypotheses

Utilizing network predictive mechanisms and a multicommodity flow algorithm facilitates improved management of network information streams. The improved management of these network streams enables the user to give better quality of service to specific information by categorizing it into levels of priority.

Research Objectives

The adaptive routing algorithm has the following research objectives:

1. Develop a priority aware routing protocol for network flows
2. Improve the quality of service for higher priority flows in the network
3. Integrate the prediction of queue sizes into the routing protocol

Research Focus

To investigate the feasibility of combining network state predictive mechanisms and a routing algorithm that balances the network bandwidth across multiple paths in the network.

Investigative Questions

The investigative question that will be looked at during this research will be whether the Network Tasking Order (NTO) can increase the QoS experienced by flows in a network and if predicting a queues utilization rate can increase the QoS experienced by flows in a network.

Assumptions

This paragraph covers the assumptions associated with the research. Information in a military network would have to be categorized into different buckets such as mission essential and routine/normal. Each bucket can then receive a priority level giving the information contained within it the same priority. This gives all flows in the network a discernible marker that can be used to prioritize the flow of information. By marking information, routing algorithms can be tailored to meet the demands of the user by enforcing prioritization of flows in the network.

Implications

If this research proves successful, the implications are that a network can be flow aware and assist the user in better controlling how information flows through their network. Another implication results in better control over the QoS experienced by flows based on a priority structure. A lower packet loss rate would mean that less traffic has to be resent thereby reducing the load placed on the network. A smaller delay would result in faster delivery of information.

Summary

This chapter introduced the problem that demand for bandwidth may outpace its availability on the GIG causing undesirable affects concerning QoS. A definition for QoS was provided and shown how it relates to the problem and solution. The internet layers were introduced and there correlation with the research shown. Priority levels and the role they play in this effort by allowing two different flows having the same source and destination could take different paths based on their priorities. This chapter also discussed the general issue and problem being investigated. Research objectives and focus were presented that will help to prove the hypothesis followed by investigative questions, assumptions, limitations and implications that deal with this research.

The remainder of the document flows in the following manner. Chapter Two (II) gives a review of other research efforts that have tried to institute a network layer protocol that give QoS guarantees with discussion on how they are different. The pieces to the Adaptive Routing Algorithm for Priority are outlined. Chapter Three (III) covers the methodology used that covers such things as approach, system workload, performance metrics and lastly the simulation setup is covered. Chapter Four (IV) contains the results of the simulations and a discussion on the investigative questions that were asked in this chapter. Chapter Five (V) the final chapter concludes the research through a discussion on the significance of the research and recommendations for future research in this area.

II. Literature Review

Chapter Overview

Chapter II starts by covering other concepts that improve the Quality of Service (QoS) of network flows through enhancement of network layer protocols. The basic operation of those concepts and some of their limitations is discussed. Followed by, how this research addresses these limitations. The remainder of the chapter is a summary of prior research efforts that the Adaptive Routing Algorithm for Priority (ARAP) flows utilizes. It covers the basic operation and ideas that those research areas cover and the type of information that the research provides.

Flow Aware Network

The Flow Aware Network (FAN) concept was introduced by [20], which provides a way for users to control traffic in a network based on what its creators call implicit admission control and per-flow scheduling. The authors state that FAN provides adequate QoS guarantees for streaming and elastic flows and it does this without class distinction or control signals to route traffic specification. An elastic flow is described as a file that is being transferred that can withstand varying transfer rates. Video or voice type flows represent streams and they typically cannot withstand varying transfer rates.

The basic pieces that make up the FAN architecture are Admission Control, Protected Flow List, Priority Fair Queuing and Cross-protect Router. The admission control block controls the start of new flows going through the router. When the system is not experiencing congestion and a new flow arrives, the ID of the flow is placed in the protected flow list, which stores all the flows currently in progress. The admission control uses a timeout parameter to let it know when to remove a flow from the list. When the arrival time between packets for a flow exceeds this parameter, the admission control assumes that the flow has completed. When congestion is detected, no new flows are allowed to start. In ARAP, new high priority flows would be allowed to start during times of congestion and low priority flows could start if there was a path from source to destination that was not congested.

The fundamental component of FAN is the cross-protect router, which is developed in [15]. This special router enables additional storage and processing of information for the system. The cross-protect router also contains a scheduler which estimates the max rate that can be realized by an elastic flow. Additionally, the scheduler is responsible for detecting congestion on the link. Congestion is present on the link when the *fair_rate<min_fair_rate* or *priority_load>max_priority_load*.

In [6] FAN was compared to other QoS architectures, such as Differentiated Services and Integrated Services, and was found to be easier to implement and also conformed to net neutrality paradigms. However, some drawbacks were noted. First, elastic flows had the potential to be broken each time the protected flow list was flushed. Second, the admission control would accept too many new flows into the protected flow list after a flush had occurred. The consequence of the latter issue was the re-emergence of congestion. To fix these drawbacks three new mechanisms were proposed in [6]. The ARAP system also prevents these drawbacks mainly because it does not utilize the protected flow list because low priority flows are expendable.

The FAN concept is similar to this research in that they are both placed within the network layer and the goals of the two ideas are to give QoS guarantees to flows based on a flow type. This research looks at priority of a flow as the discriminator for QoS whereas the FAN concept uses streaming and elastic flow types as its discriminator. In the ARAP system, the high priority flow is the protected flow on a global scale unlike the FAN that has a protected flow list on each element. Having a separate protected flow list on each element could potentially cause a problem if a flow is considered protected on one router but not another. The ARAP system does not suffer from the same affect. Finally, the FAN system still relies on Transmission Control Protocol (TCP) to help establish the correct transmission rates. ARAP does not rely on TCP to ensure that transmission rates are constant.

Multiprotocol Label Switching

Multiprotocol Label Switching (MPLS) is another form of traffic engineering in which an application exploits alternate routes between source-destination pairs to balance the congestion in the network. ARAP and MPLS are both implemented at the router level and they share the same goal of increasing the QoS through balancing the load across multiple paths in the system. A couple of differences between the two are that MPLS attaches a packet header to the packet making each packet a little larger, which increases the needed bandwidth for a flow. This has an adverse effect on links that are considered to have a low bandwidth or currently experiencing congestion. If ARAP were to be deployed in the real world than it could keep the same standard internet protocol header unlike MPLS, which has to create a new packet header that, it places over top of the existing information. For example, the ARAP would utilize the Traffic Class field for IPv6 type packets and the Differentiated Services field on an IPv4 packet.

MPLS employs a special router referred to as a label switch router [27]. Label switch routers are located at the edge of the network and they encapsulate incoming packets with a new header and remove the header just before a packet leaves the network. Subsequent routers refer to this label to ensure the packet goes out the correct outgoing link.

The information contained in the internet protocol header of the packet and local network information is the basis for the MPLS label that is attached to a packet as it enters the MPLS network. The interior label switch routers inspect the labels on the incoming packets then send them to the appropriate outgoing link and replace the current label with a new one as required.

In [7] the approach is to try to balance the traffic bandwidth on multiple label switched paths between the ingress and egress nodes. The balancing of traffic in [7] is accomplished through a MPLS Adaptive Traffic Engineering technique. MPLS Adaptive Traffic Engineering technique uses a dual phased approach that includes monitoring and load balancing phases.

The monitoring phase measures packet delay and loss via probe packets. To do this, the system sends out probe packets from the ingress node to an egress node based on the traffic class they are monitoring. The egress node will then send the packet back to the ingress node with information that will allow it to calculate the one-way trip time and packet loss rate. When the monitoring phase detects congestion in the link it switches to the load-balancing phase.

During the load-balancing phase, the traffic-engineering block makes decisions about which flows need to be changed to equalize the traffic on the congested label switched paths. In other words, if the delay or packet loss rate is high for a particular label switched path the traffic-engineering block will make changes to incoming flows so that the flows can avoid the congested paths.

The MPLS networks in [27, 21] utilize the current state of the network to calculate the label switched paths and to balance the network bandwidth across those paths. When [7] monitors the demand on the label switched paths they use real time measurements to make decisions. Currently, MPLS does not look at using a predicted network approach however, that is possible using a Network Tasking Order (NTO) and Network Weatherman (NWM).

Network Tasking Order

A NTO is a concept explored by Matt Compton. The Air Force does not currently utilize this concept as presented in [4]. However, network routing algorithms can be developed that utilize the types of information provided by a NTO. The NTO concept provides a snapshot of what the network will look like in the future and "directs the day-to-day operation of specific portions of the GIG" [4]. This advanced knowledge of network state will enable a routing algorithm to preplan routes for traffic flows.

Information Provided

The NTO contains a vast amount of daily information about the GIG. Much of that information, as it pertains to specific networks on the GIG, can be pulled out and utilized to create efficient routes for information flows in the network. The information provided by the NTO includes such things as when and where additional potential nodes will be located, what kind of service they can provide to the network, and what types of connections they can support. This research builds on this idea by saying that the NTO also provides information on potential source-destination pairs that share high priority information throughout the course of a day.

Network Weatherman

The NWM is a stochastic estimator based on a Kalman Filter design that enables the prediction of future queue sizes for specified queues in a network [24]. A limitation that the NWM contains is that it must be tuned to the network for it to sufficiently predict future queue sizes. Tuning of the NWM is accomplished by finding values for the variables that represent the variance of the dynamic noises given in [24].

With the knowledge of a potential future state of network routers, a routing algorithm has the ability to make advanced decisions that could increase the QoS at the network layer. For instance if it is predicted that a specific router will become full at some time in the future the routing algorithm can alter some flows and send them along another path.

Information Provided

The NWM provides a potential future state of the network in the form of predicted queue sizes. The NWM provides predictions on an interval basis that is controlled by an external variable and this variable can be changed during operation. Additionally, the user can set how far into the future they want the predictions to take place however, this value is set at the beginning of the simulation.

Multicommodity Flow Algorithm

The Multicommodity Flow (MCF) algorithm is a family of algorithms that attempts to send as much information as possible across a network given some constraints and an objective function. Three commonly used MCF algorithms are the Max Concurrent Flow, Maximum Multicommodity Flow and Minimum Cost Concurrent Flow [10, 12, 13, 24]. The next few sections discusses the general description of a multicommodity flow problem followed by the maximum concurrent flow problem which is used to route higher priority flows for this research effort.

Multicommodity Flow Problem Description

A Multicommodity Flow problem is defined using the following nomenclature. A directed network G with a set of vertices and edges called V and E. Each edge in the network has a corresponding capacity u. In addition, there are multiple source destination pairs contained respectively in sets labeled as S and D. Individual source destination pairs are $(s_i, d_i)$where $1 \leq i \leq k$. The value k is the number of source destination pairs in the system. The problem is to route the flows f_i through the system from $s_i$to $d_i$that satisfy some node conversion constraints as well as to meet an objective function criterion without exceeding the edge capacities in the graph, such that the sum of all the flows going over a particular edge does not exceed its capacity [10]. The following more completely describes the mathematical multicommodity flow problem from: "A multicommodity flow problem is defined on a directed network $G = (V, E)$ with capacities $u : E \rightarrow \boldsymbol{R}$and $k$source-sink terminal pairs$(s_i, d_i), 1 \leq i \leq k$.[10]"

Criterion:

Equation 1

$$\forall e\colon \sum_{i=0}^{j} j_i \leq u(e)$$

Maximum Concurrent Flow Problem

A Maximum Concurrent Flow Problem (MCFP) is a subset of the Multicommodity Flow problem. The MCFP is where source-destination pairs can send and receive information concurrently. The throughput ratio between all flow supplied by the (s_i, d_i) pairs must be the same [24]. More specifically each flow j has assigned to it a demand d_j where the objective is to maximize the ratios of all demands given by the following objective function for a MCFP [10]:

Equation 2

$$\max \lambda, \left|f_j\right| \geq \lambda d_j, \forall j$$

In essence, this is saying that all flows will receive the same bandwidth ratio, based on the flow that is the limiting factor for the group.

The downside to this approach is that no one flow can send its entire throughput unless there is room for all flows. The causal effect is that all flows are treated equally and that hinders one's ability to use priority as a qualifier for routing. My research uses the idea of the MCFP presented by [10] but does not limit a flow's throughput based on the demand for one particular flow. The objective of my research is to maximize the amount of information flowing across the network while at the same time providing quality of service guarantees to higher priority information.

Fleischer has been able to construct a multicommodity flow algorithm that is faster when k > m/n, or more specifically, when the number of commodities k is divided by m edges over n nodes.

1 **Input:** network G, capacities $u(e)$, vertex pairs (si,ti) with demands d_i, $1 \leq i \leq k$, accuracy ϵ
2 **Output:** primal (infeasible) and dual solutions $x$and l
3   Initialize $l(e) = \delta / u(e)\ \forall e, x \equiv 0$.
4   **while** $D(l) < 1$
5   ***for*** $j = 1$ *to* k ***do***
6         $d'_j \leftarrow d_j$
7   **while** $D(l) < 1$ and $d'_j > 0$
8             $P \leftarrow$ *shortest path in* $\mathcal{P}_j$ using l
9             $u \leftarrow \min\{d'_j, min_{e \in P}\, u(e)\}$
10             $d'_j \leftarrow d'_j - u$
11             $x(P) \leftarrow x(P) + u$
12             $\forall e \in P, l(e) \leftarrow l(e)\left(1 + \frac{\epsilon u}{u(e)}\right)$
13   **end while**
14   **end while**
15 **Return** (x,l)

Figure 1: Multicommodity Algorithm [10]

Figure 1 depicts the algorithm from Fleischer that is used in this research to spread the flow out for the source destination pairs. The inputs and output of the algorithm are listed in lines 1 and 2. Line 3 starts the algorithm where all the edges in the graph are initialized to a length of delta divided by the capacity for that edge. This initialization step makes the edges with the larger edge capacity more favorable to the shortest path algorithm. Line 4 checks the termination condition. $D(l)$ is calculated by multiplying all edge lengths by their respective capacities and summing them up which, is then compared to one. Line 6 takes makes a copy of the demand for a particular commodity. Next, in line seven is the termination criterion for the inner loop is checked. Line 8 finds the shortest path using Dykstra's shortest path algorithm while at the same time checking to ensure the demand for the commodity can be satisfies by all edges in the path. Lines 9 and 10 are used when trying to find the true max concurrent flow where commodities can be split up on multiple paths. This research is not using splittable-paths; therefore, these two lines are ignored. Line 11 adds the path to the set x and the associated demand for that commodity. Line 12, then, lengthens each edge in the path by a small amount. The small amount is described as epsilon multiplied by the demand required by the commodity divided by the capacity for that edge. The lengthening of the edges in the path prevents overuse of any particular edge in the system.

The value chosen for epsilon directly affects the runtime of the algorithm. The value of delta affects only the starting edge lengths for the algorithm but if a small enough delta is not used then the sum of all edge lengths times their capacity could cause the algorithm to not enter the first while loop by being greater than one at the start of the algorithm.

A randomized rounding algorithm is then used to take the output from the Fleischer algorithm to then choose paths in the network to route the flows without violating any of the edge constraints.

Summary

Chapter II discussed other research efforts that made enhancements to the network layer such as the FAN and the MPLS. Both have shown to improve the QoS in the networks that they are implemented on as well as balance out the bandwidth demand across the network edges. Some of their limitations were that special routers would be needed to implement in the case of the FAN or additional bandwidth was being used because of the need to implement a new header that only worked in that network in the case of MPLS routing. One key aspect of both these approaches is that they rely on the current state of the network in order to make their adjustments.

The last part of the chapter discussed the research associated with the parts of the ARAP. This research included the NTO, NWM and multicommodity flow algorithm, the key aspects of this research were covered along with the information that each provides to the ARAP. The subsequent chapter goes into the methodology behind the ARAP research and covers how the research covered in last part of this chapter goes into the making of the ARAP.

III. Methodology

Chapter Overview

This chapter discusses the methodology to evaluate the Adaptive Routing of Priority Flows in a network. This chapter is organized in the following manner. First, the approach for the overall research area is discussed, which covers how the logic behind the Adaptive Routing Algorithm for Priority. Additionally, the approach provides an overview of the research being conducted and describes the scope of this research. Second, the problem is defined, which includes the goals and hypothesis of the research. This section also covers the approach to the experiment and how the stated goals are achieved. Third, system boundaries and various system attributes are covered to include services, workload, performance metrics, system parameters and factors. Finally, the evaluation technique and the experimental design are covered followed by a summary of the chapter's main points.

Approach

Network flows are managed through various mechanisms including multicommodity flow algorithm, caching scheme, Network Tasking Orders (NTO) and Network Weatherman (NWM). The multicommodity flow algorithm used to set up the routes for each of the high priority flows is from [10] which has a runtime of O($\epsilon^{-2}m(m + k)log^{O(1)}m$) where m is the number of links in the network, k is the number of commodities in the network and ϵ is the desired accuracy of the solution.

The NTO provides advance knowledge of network behavior and assigns priorities to each of the flows. The NTO contains additional node and link information over and above normal topological information of the network. A near-term estimate (or snapshot) of network conditions is given by the Network Weather Man (NWM) which provides an estimate of the future queue size for a given node. The NWM is used on the most heavily used links to predict queue sizes. Another agent uses the predicted queue size to restrict packets allowed on that link by giving higher priority packets access to the link while making the lower priority flows find another way to their destination. The caching scheme is the initial rerouting mechanism for the lower priority flows. If the cached route for the flow is also unavailable then the agent will try to find another route for the lower priority flow if the agent cannot then the flow is stopped.

Adaptive Routing Algorithm for Priority

The Adaptive Routing Algorithm for Priority (ARAP) takes input from the NWM, NTO and the network topology. This information is an input to the adaptive routing algorithm (as seen in Figure 2) and produces the routing tables that are then installed on the routers in the network.

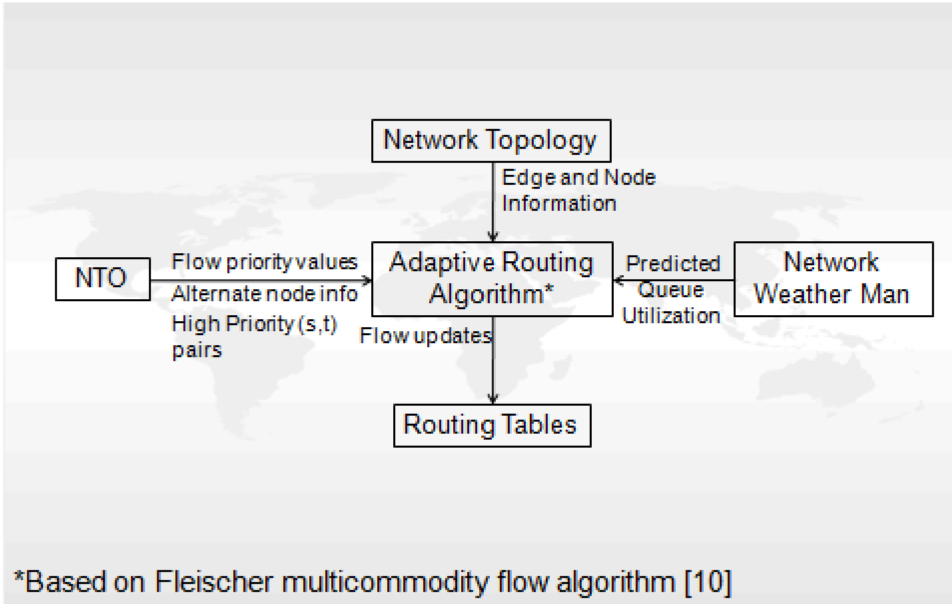


Figure 2: Adaptive Routing Algorithm Vision

The NWM components make a prediction every 0.5 seconds and they predict out six tenths of a second into the future. For this research, the predicted information is sent out of band however, for a real life network, this information would travel over the same edges as the network traffic is using. This could cause additional side effects not explored in this research and will be discussed in the future work section.

Routing Algorithms Used

The ARAP design is based on a series of external inputs that triggers various mechanisms to calculate and trigger the installation of network routing tables. There are two types of routing algorithms that make up the ARAP the first is the multicommodity flow algorithm that was discussed in Chapter II that handles the routing of the high priority flows and the Floyd Warshall algorithm calculates the shortest path routes between each node and it utilized by the low priority flows.

This seems counter intuitive at first glance because one would expect that the higher priority flows would utilize the shortest path. However, the routing algorithms were chosen for these priority levels for the following reasons. First, in order to utilize the multicommodity flow algorithm discussed in chapter II prior knowledge of the flow information is needed. The NTO provides this information but only for the high priority flows. As a result, routes could not be precalculated for all low priority flows without utilizing shortest path routing since any possible node can send to any other node. Second, low priority flows did not need to be spread out over the network because if an edge were congested the low priority flow would be rerouted around the congested link.

The NTO nodes are only utilized by the low priority flows because they are sent best effort. High priority flows do not utilize the NTO information to prevent possible disruption of those flows due to the potential unavailability of the NTO nodes.

How the Adaptive Routing Algorithm Works

The algorithm takes in the network topology information as well as the information for the high priority flows given by the NTO and the multicommodity flow algorithm is used to calculate the routes for the high priority flows. This algorithm returns multiple possible paths if they exist for each flow and a randomized rounding algorithm is used to pick which route to take. Then routing tables for those flows were installed. Next, the Floyd Warshall algorithm is ran twice once without the NTO information and once with the NTO information and the second time it is ran the routing table information is cached waiting for the NTO nodes to become available.

The system checks every two tenths of a second to see if the NTO nodes and edges are available. When they are available, the system would install the routing tables that used the NTO information. When the NTO nodes and edges are no longer available, the network switches back to the routing tables that do not utilize the NTO information.

If the ARAP receives a predicted queue size value that is above the queue utilization parameter it will hide that edge from the network and the Floyd Warshall algorithm is ran again. When the utilization of a queue drops back below twenty percent that edge is placed back in the network for low priority flows use.

Simulation Setup

Software and Operating System Details

The simulation is set up on a Linux computer system running Centos 5.8 with Kernel version 2.6.18-308.1.1.el5. The simulation is run with ns-allinone-2.34 that includes added functionality developed at AFIT. The agent developed for this simulation utilizes the code base from Captain Larry Llewellyn with some significant modifications. The use of a MATLAB2010b engine is necessary for the incorporation of the NWM. However, NWM was created using MATLAB2007b therefore in order to get NWM properly integrated into the simulation the MATLAB2007b libraries are used at compile time.

Network Setup

A software topology generator developed by Georgia Tech is utilized to generate the network topologies used and it is referred to as GT-ITM. The GT-ITM transit stub routine was used to generate the four topologies. Topologies 1 and 2 are shown in Figures 3 and 4 respectively. Topology 3 and 4 are located in the Appendix A. Figures 3 and 4 portray a group of small nodes that are connected to each other through single links.

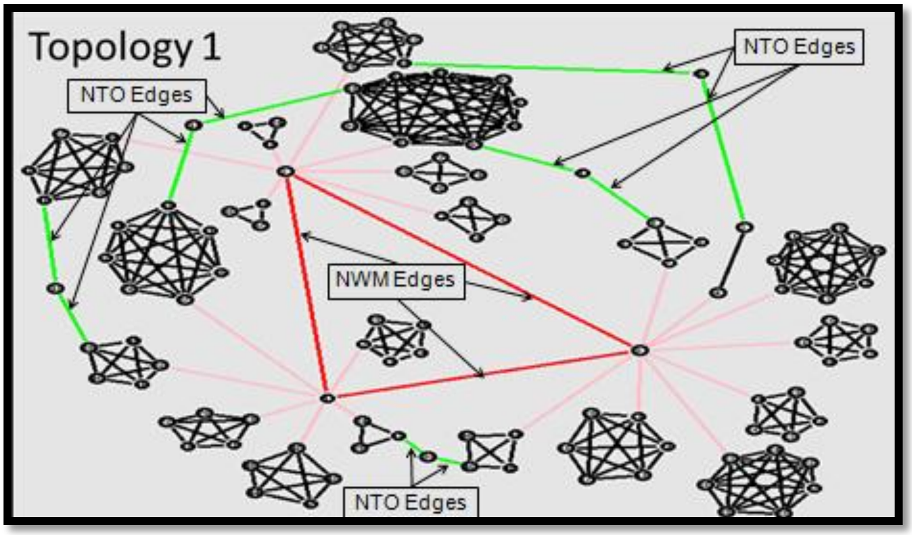


Figure 3: Topology 1

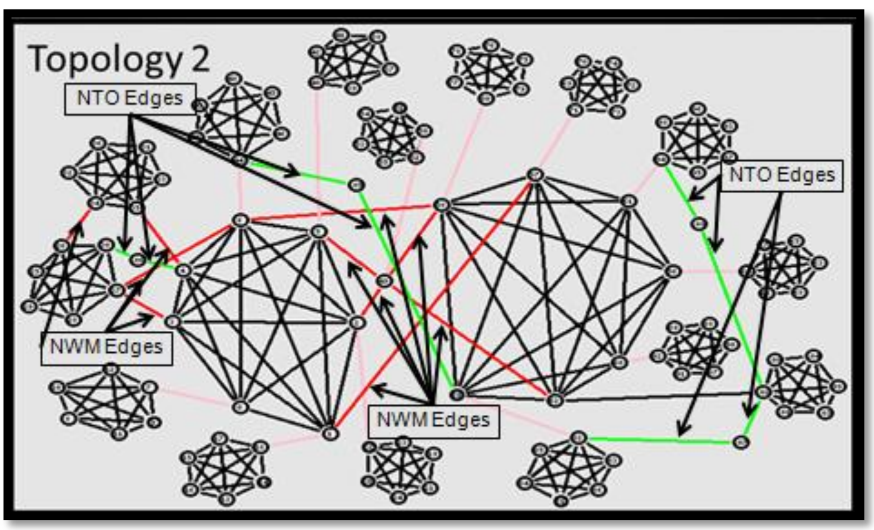


Figure 4: Topology 2

The Table 2 shows the parameter values used to generate topologies 2 and 4. The values in Table 2 created a network with 105 nodes and approximately 580 one-way edges.

Table 2: GT-ITM Variables Used

Number of transit domains	1
Average nodes/transit domain	3
Average stub domain/transit node	5
Average nodes/stub domain	6

Appendix B shows the file used by the GT-ITM program to produce topologies 2 and 4. In addition to the variables listed in Table 2 Appendix B shows some other variables that affect how the topologies are constructed.

Flow Generation

A random scheme is used for the generation of flows for the simulation. NS-2 is partially built using Tool Command Language (Tcl) which has built in random number generators that were used to generate the random flows for the network. A flow consists of a Source and Destination node, Start Time, Priority Level and Size. Table 3 displays which distribution is used for each part of the flow listed above.

Table 3: Types of Random Distributions used for Flow Generation

Part of Flow	Distribution Type
Flow Source	Uniform
Flow Destination	Uniform
Start Time	Exponential
Priority Level	Uniform
Flow Size	ParetoII

An equal likely hood of being chosen was need for source, destination and priority level parts of a flow therefore a uniform distribution is chosen. Internet traffic is considered to have a heavy-tailed distribution [11]. The ParetoII distribution in Tcl is chosen to represent the flow size because it provides the heavy-tail distribution needed.

Finally, an exponential start time is needed to mimic the exponential arrival of packets at each of the nodes. Table 3 does not show a stop time for the flows because it was calculated based on the start time and the flow size.

The number of flows generated at the beginning of the simulation depends on the total bandwidth of the network and the bandwidth demand variable found in Table A. For example if the if a network has 20 edges and each edge has a bandwidth of 2 MB with the bandwidth demand variable set to 0.65 than 26 MB worth of flows will be generated.

Flow Routing

The NTO gives source destination pairs for high priority flows therefore the maximum concurrent flow algorithm developed by Fleischer is used to route the high priority flows priority at the start of the simulation. This more evenly distributes the higher priority flows around the network to help prevent one or more links from being over utilized. The lower priority flows utilize a shortest path route based on the source and destination node.

System Boundaries

The ARAP system includes the network, that consists of nodes and links, the flow agent, NWM, and the NTO. The nodes in the network are responsible for routing the flows through the network according to their routing table. Links in the network carry the flows from one node to another. The link delay is not considered as a part in this system and is set to 15ms for all edges in every simulation. The flow agent as part of the system is being tested and compared to other simulations that do not utilize an adaptive flow agent. The flow agent ensures that the higher priority flows get preference on congested links between clusters of nodes using information from the NWM and NTO. Figure 5 shows a notional system diagram.

The NWM sends the flow agent updates on predicted queue sizes while the NTO gives advance notice of the expected state of the network up to 24 hours in advance. The advance notice includes node and link information, as well as guidelines for assigning priorities to flows in the network. Figure 6 shows the system components and the inputs and outputs of the system.

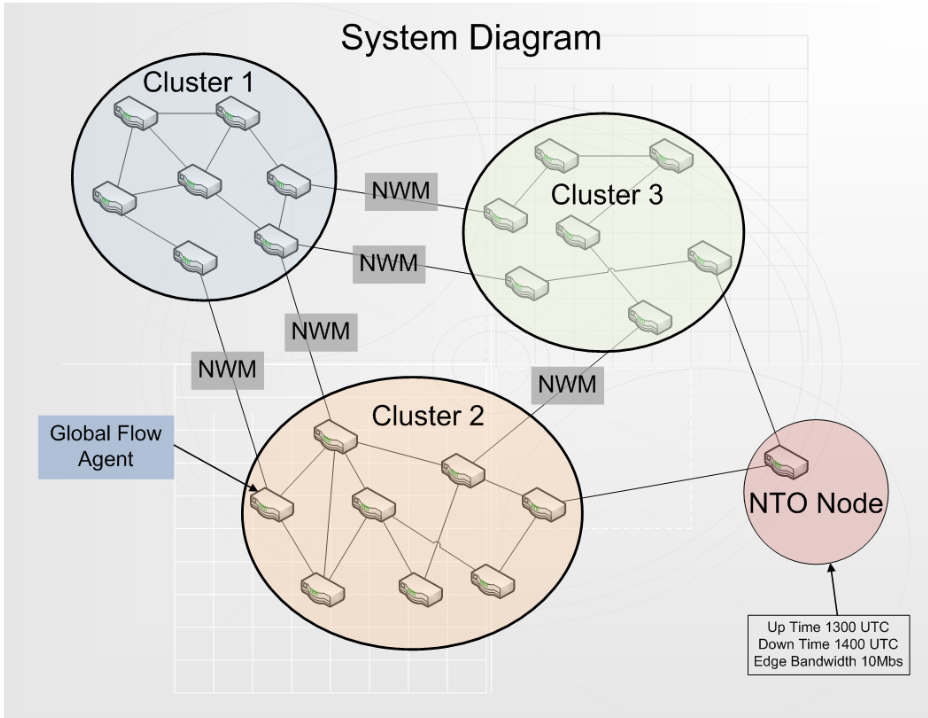


Figure 5: Notional System Diagram

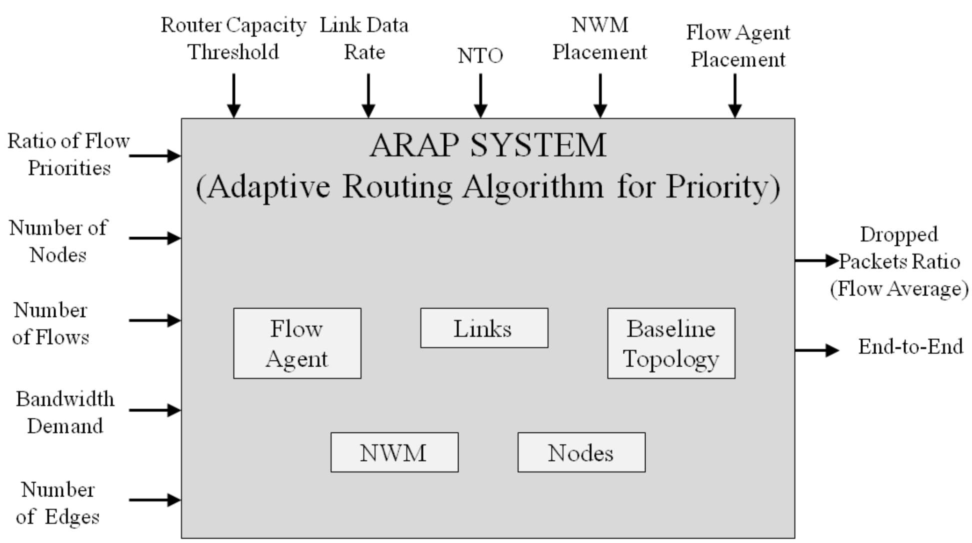


Figure 6: System Parameters

System Services

The creation of routing tables is the service that the Network Priority Flow Optimization System provides. This service ensures that higher priority flows receive better quality of service in the network. The network routing tables contain the outgoing link that a packet in the flow will go through based on a specific flow priority and the corresponding destination. To provide this service, there are two subservices:

- A prediction of the queue size for specific nodes in network
- Assignment of priority values to flows at the flow source node based on NTO

The NWM sends periodic updates to a Flow Agent in the form of predicted queue size at a specific router. The assignment of priority flows is based on the importance of the information being sent. The NTO contains the classification levels for the type of information in the network.

Outcomes for these services are:

1. Routing tables provide better quality of service to higher priority flows.
    a. Lower priority flows see an increase in quality of service.
    b. Lower priority flows see only a minor degradation in service.
    c. Lower priority flows see a major degradation in service.
2. Routing tables do not provide better quality of service to higher priority flows or it is worse.
    a. Lower priority flows see an increase in quality of service.
    b. Lower priority flows see only a minor degradation in service.
    c. Lower priority flows see a major degradation in service.
3. Predicted queue size is either correct or not correct.

4. Assignment of priority levels is either correct or not correct.

This research is only interested in outcomes 1, 2 and 3. That is, it is assumed that the priority levels are correctly assigned at the source node of the flow.

Workload

The overall workload of the system is a function of the configuration of the network. This configuration is dependent upon the number of nodes in the network and the number of links connecting the nodes. The environment in which the system operates is affected when the workload parameters of the system are changed. The workload parameters are discussed below.

Bandwidth Demand

The bandwidth of a flow includes both the size of an individual packet and the rate at which the source node sends a series of packets. The bandwidth demand on the network is dependent upon the number of flows in the network.

Number of Nodes and Links

The number of nodes and links affect how long it takes the algorithm to calculate the paths for flows in the network. The more nodes and links there are in the network the longer it will take the algorithm to run.

Ratio of Flow Priorities

This workload parameter affects how well the flow agent functions. As the ratio of high to low priority flows increase, the workload on the flow agent also increases.

Number of Flows

The number of flows affects the runtime of the algorithm. If there are more flows to be route then it will take longer to compute the routing tables.

Performance Metrics

The performance metrics are the attributes of the system that are measured to determine if the system is meeting its goals. The following paragraphs describe the performance metrics and how they are measured.

Dropped Packets Ratio per Flow

One of the goals of the research is to ensure that higher priority flows experience lower packet loss than the lower priority flows. This metric is used to measure the ratio of dropped packets to total packets sent in that flow. It categorizes each flow by its associated priority, which facilitates comparison of dropped packets based on priority. A dropped packet is counted when a queue is unable to forward the packet. The unit of this metric is the number of dropped packets in a flow over the total packets sent in that flow.

End-To-End Delay per Flow

The end-to-end delay of a flow is measured from the time the packet leaves the source until the time that the entire packet has been received at its destination. The end-to-end delays are categorized by the priority assigned to the flow. The unit of this metric is milliseconds.

System Parameters

A system parameter is an attribute of the system that if varied will affect the response of the system. The system parameters are discussed in the following paragraphs.

Network Weather Man Placement

The NWM component is placed on links that are expected to have higher congestion. This placement provides the system better visibility into the current state of the network. If the NWM component is placed on links with less congestion, the system may not ever get a recalculate message because the queue may not reach the threshold value.

Router Capacity Threshold

The Flow Agent uses the router capacity threshold. When the threshold value is exceeded the Flow Agent recalculates the routing tables for the network to reduce utilization of that router by lower priority flows. A lower threshold value causes a higher workload on the system due to more frequent recalculations.

Network Tasking Order

The NTO supplies the Flow Agent with advance notice of network state. The correctness of the state information can affect the response of the system. Having some invalid future state of the network causes the system to have a higher workload since it is unable to precalculate the routing information.

Link Data Rates

The link data rate is the capacity of a link to carry data measured in Mbps. An increase in this rate causes additional workload on the system in the form of increased queue sizes and more rerouting of lower priority flows.

Flow Agent Placement

The flow agent placement specifies where flow agent is located inside the network. There currently is only one flow agent and the placement inside the network is arbitrary.

Factors

Table 4 lists the factors for this research and the corresponding levels for each. The following paragraphs describe the factors selected from the preceding parameters. How the factors are varied and to what extent are discussed.

Table 4: Experimental Factors

Ratio of Priority Levels High to Low	4:1	1:1
Bandwidth Demand, Percentage of Network Bandwidth	~65%	~40%
Routing Table Update Threshold	50%	70%
Network Tasking Order Validity	High	Low

Ratio of Priority Levels

A flow is assigned one of two priority levels: high and low. The ratio is expressed as high to low and the corresponding levels are: 4:1 and 1:1. These levels are chosen to determine how the system reacts when there are many higher priority flows compared to lower priority flows in the system. As the ratio of higher to lower priority flows increase, the system should experience a higher workload as it tries not to drop any packets from the higher priority flows.

Bandwidth Demand

The system experiences two different kinds of bandwidth demand: high and normal. High bandwidth demand is defined as approximately 65 percent of the total bandwidth of the network. For example if the total bandwidth of the network is 100 GB per second then total demand from all the flows is set to 65 GB per second. A normal demand is defined as approximately 40 percent of the total bandwidth of the network. These levels are defined to ensure that the system reacts in an expected manner. That is, as the bandwidth demand increases the higher priority flows receive priority placement in the queues. It is expected that the lower priority flows will experience a higher rate of packet loss than the higher priority flows.

Routing Table Update Threshold

The routing table update threshold has two levels: 50 and 70 percent full. This threshold value is tied to queue utilization. The two threshold levels evaluate the time it takes the system to react to the predicted queue size. It is expected that as the threshold increases, the system will experience an increased number of packets lost in higher priority flows.

Network Tasking Order

The NTO correctness levels are high and low. When the level of correctness is set to high, the network state will be exactly as the NTO states. When the level of correctness is low, the network state is not as the NTO predicts and the normal network topology is used.

These two correctness levels are chosen because missions in the military can change rapidly, therefore the Flow Agent must continue to provide valid routing tables for the nodes even when the network state does not match that of the NTO. When the NTO is not correct, the preferred service to higher priority flows cannot be guaranteed and all flows will experience similar delay and packet loss.

The simulation is run for 100 seconds and when the NTO is valid, the nodes and edges will be available for use by the low priority flows during the following times, 20 to 40 seconds and 60 to 90 seconds. The number of NTO nodes that each system was able to utilize was 5 extra nodes and 10 additional edges that connected the nodes to the existing graphs.

Evaluation Technique

A network simulation is used to evaluate the quality of service for network flows with varying levels of priority. The network simulation environment is created using Network Simulator 2 (NS2), a widely used network simulator. In addition to the network environment, the routing protocol for the nodes in the network is developed in C++. Simulation was chosen because it is easier and cheaper to build a simulation with the infrastructure needed for the experiment than using other methods. In addition, the parameters that are being varied are easier to control in a simulation environment.

A flow agent is created using the C++ programming language and is inserted into NS2 framework to control the routing tables of the nodes in the system. The flow agent takes input from the network, the NTO and the NWM. The NWM components are placed on the links that connect the different clusters in the network and any link that is thought to have high utilization. The flow agent calculates the path taken for each flow in the network. This path is based on queue size threshold value, the priorities assigned to each flow by the NTO and the predicted queue sizes for the nodes provided by the NWM. Each time a queue size threshold value is reached for a node, the Flow Agent recalculates the routing tables for the lower priority flows. Priority queues are used throughout the simulation, which means that if the queue size is exceeded all queues including queues linked to NWM begin dropping the lower priority packets. For example, if the threshold value is set at 50 percent, the recalculation of routing tables will begin only if the predicted value sent by NWM reaches this threshold.

The output file from the simulation is used to calculate the total number of packets sent by the source nodes and the total number of packets received by the destination node. The file also gives the ability to calculate the delay felt by each packet from source to destination.

Parts of the simulation can be validated using similar research conducted at the Air Force Institute of Technology. The NWM data for the Flow Agents is validated using initial NWM data [Stuckey 2007].The Flow Agent is validated using two similar simulation configurations called No Update and Queue Update that run the exact same scenarios with a few differences that will highlight the utility of the ARAP. The first configuration is called No Update and it utilized the same routing algorithms and network setup as the ARAP however, no rerouting is accomplished. No Update shows what happens in the network when nothing is done to reroute the flows due to congestion. The second configuration is called Queue Update and is the same as the ARAP design except that instead of using the predicted queue sizes the system utilizes real time queue sizes. This enables a comparison between predicted queue values and real time queue values. Table 4 in the factors section of this chapter gives us 16 different scenarios that are looked at and each scenario had 30 different runs associated with them.

Experimental Design

The experimental design scheme chosen is full factorial with a 90 percent confidence level for dropped packets and 95 percent confidence interval for flow delays. The four factors chosen each have two levels to be tested. This leads to 2x2x2x2 = 16 different experiments for the ARAP system. The confidence level is used because combining a flow control agent and queuing prediction mechanism is likely to produce higher variance in some of the metrics.

With the expectation of high variance in the system and a confidence-level of 90 percent, each experiment is repeated 30 times. Therefore, to achieve the desired confidence interval 480 total experiments are required.

Summary

The number of devices connecting to DoD networks continues to grow to include devices used in the battle space. These devices send and receive the majority of their information through the network. Therefore, it is critical that high priority information makes its way through the network with a better quality of service than low priority information. This research implements a network layer protocol for flows with a given priority.

The overall goal of the research is presented: improve the quality of service for flows with a high priority using an adaptive network routing algorithm through simulation using NS2 and TcL. The system and workload parameters for the system were described. The performance metrics described will demonstrate that the system delivers better quality of service to high priority flows. The factors varied show how the system performs without key inputs to the routing algorithm.

IV. Analysis and Results

Chapter Overview

This chapter is divided into several different sections that include component validation, primary simulation results and secondary simulation results. The component validation section covers the steps taken to ensure accuracy and legitimacy within the simulation. The primary simulation results show the results from the ARAP simulation runs and compares them to two other simulation configurations discussed in Chapter III. The secondary simulation results were completed to explore limited differences that were highlighted in the primary simulation results.

Component Validation

This section is broken down into several subsections that cover the process of validating each of the components used in the simulation.

Flow Generation

Flow generation accomplished via Network Simulator 2 (NS2) using NS2's built in random number generators. Uniform, exponential and ParetoII distributions were employed by the simulation to create the random flow profiles. With each scenario consisted of 30 runs and a different seed value was chosen for each. The same seeds were used for all three simulation configurations so that each configuration would experience the same flow generation profile for each scenario.

Uniform Distribution

The source, destination and the priority level for a given flow was determined by the uniform distribution. Figure 7 depicts the uniform distribution for the selection of the source and destination nodes for a particular scenario and run. The standard error about the mean for the number of times that a particular node is chosen is 1.54. This results in a 95 percent confidence interval of 114 to 120. Therefore, Figure 7 shows that the NS2 uniform random number generator provides a uniform distribution for the simulation scenarios.

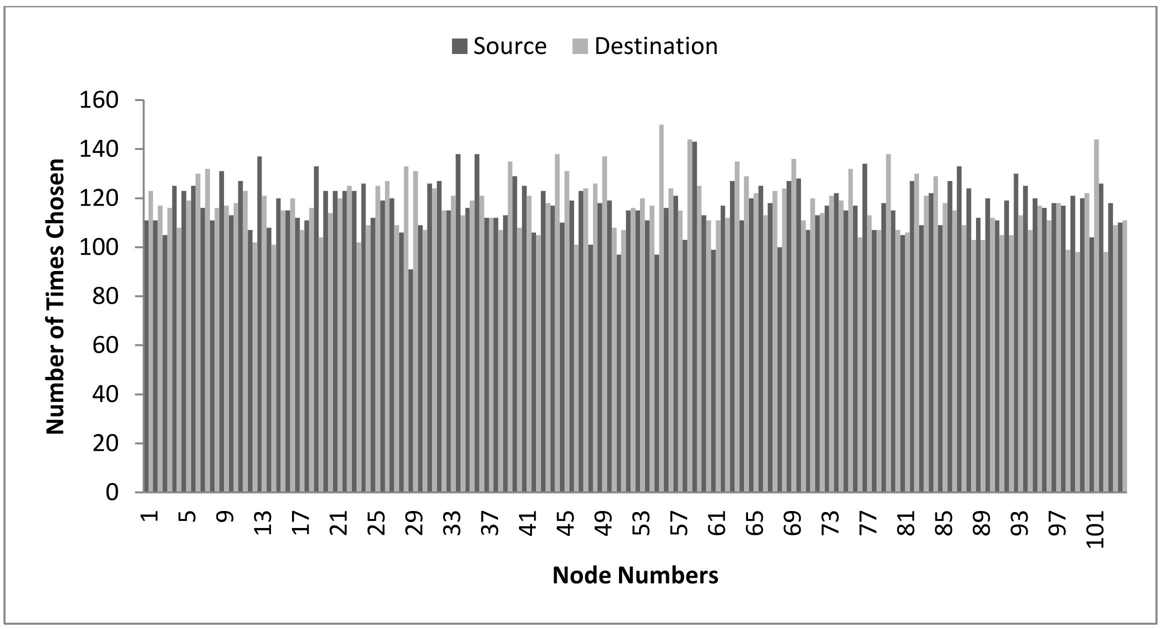


Figure 7: NS-2 Uniform Distribution for Source and Destination Nodes

Priority is also assigned based on the uniform distribution that resulted in 6023 high priority flows and 6191 low priority flows for the 1:1 ratio. The 4:1 ratio resulted in 9477 high priority flows and 2401 low priority flows. Both are within one significant digit away from being actual 1:1 and 4:1 ratios.

Exponential Distribution

An exponential distribution was used to generate start times for each of the flows to provide an exponential arrival rate of packets in the system. Figure 8 shows the start time versus the flow number, which depicts a slightly larger concentration of flows starting before 60 seconds. Figure 9 presents a better view of the start times as they are displayed in ascending order based on start time. The line in Figure 9, as depicted, shows only a slight exponential characteristic for the start time. The two figures combined show that the profile of the start times used for the simulations are random and exponential in nature.

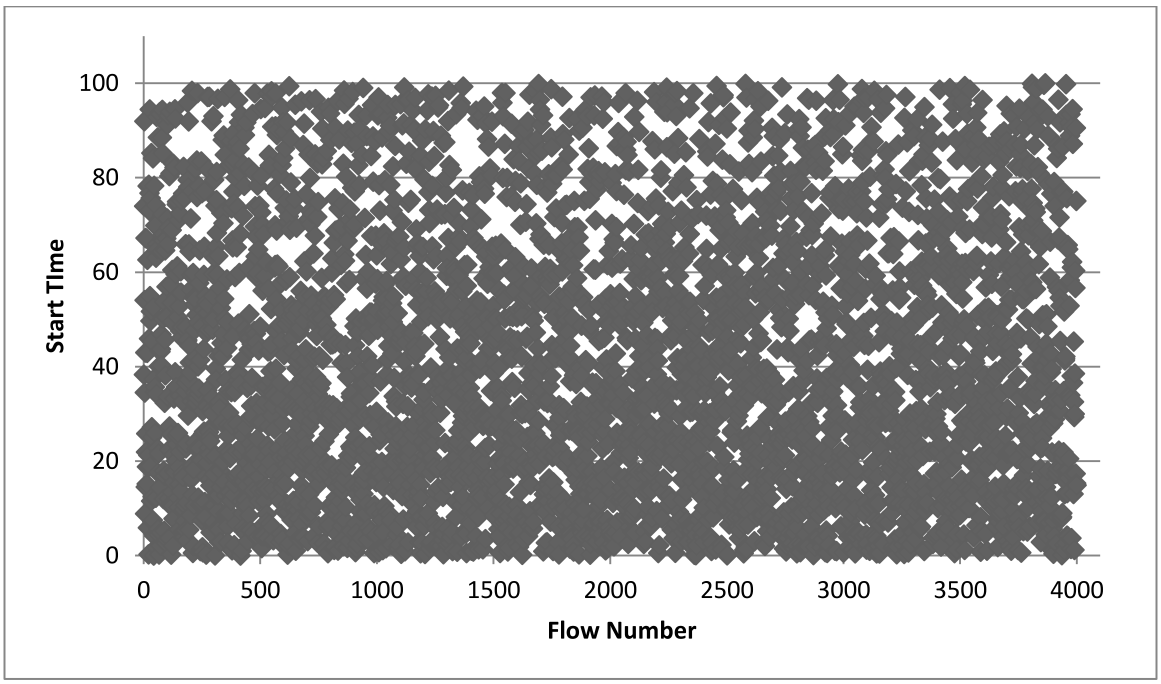


Figure 8: Start Time versus Flow Number

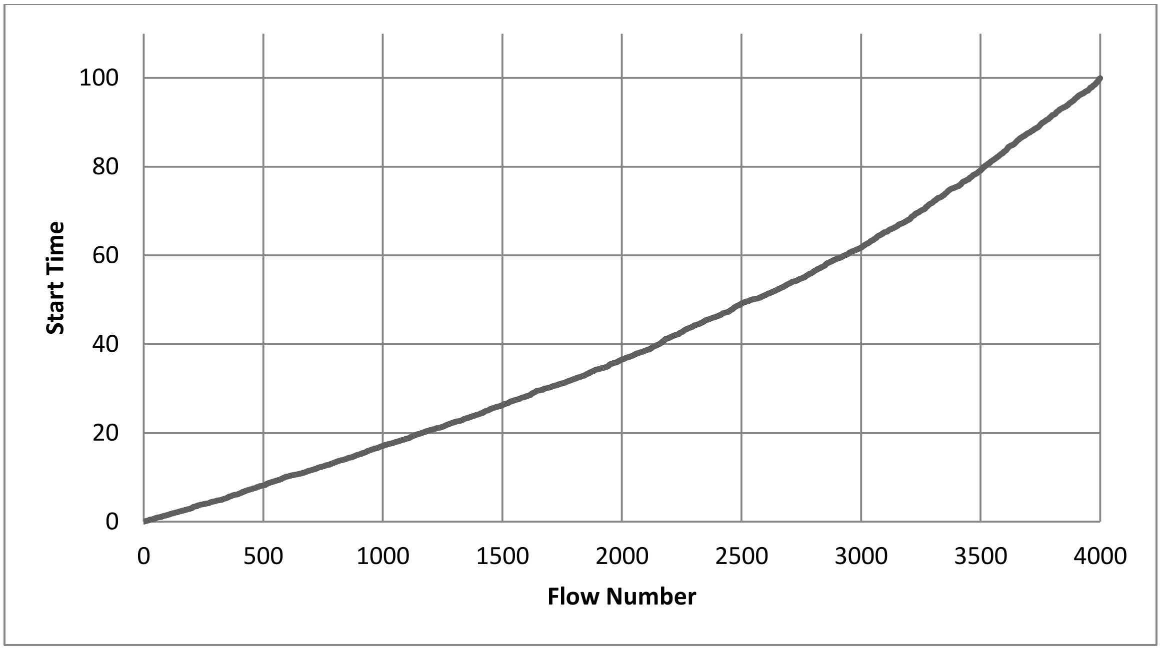


Figure 9: Flow Start Time in Ascending Order

ParetoII Distribution

Internet traffic displays a heavy tailed distribution characteristic with respect to file size and NS2's ParetoII distribution provides a way to mimic the file size characteristic for internet traffic. Figure 10 depicts a representative example of the flow sizes used for the simulations and they are shown in ascending order arranged by start time. To show that this does actually represents a heavy tailed distribution, the flows were rearranged from smallest to largest as seen in Figure 11. The combination of these two figures show that the flow profile for size is indeed heavy tailed in nature and the assignment to a particular start time is random.

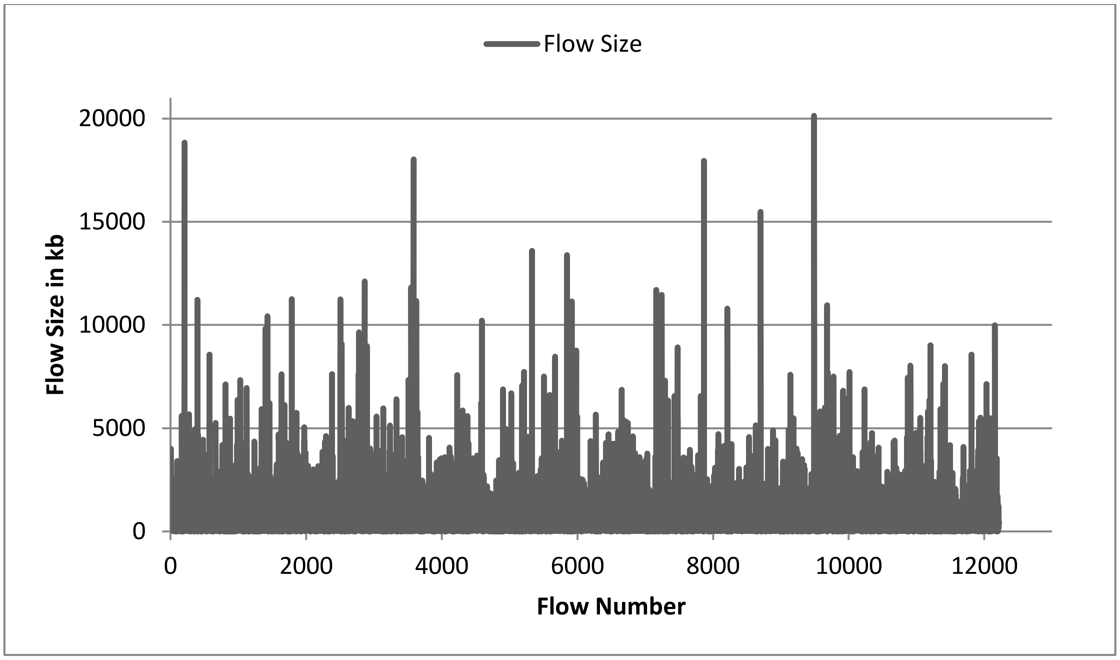


Figure 10: Flow Size Arranged by Start Time

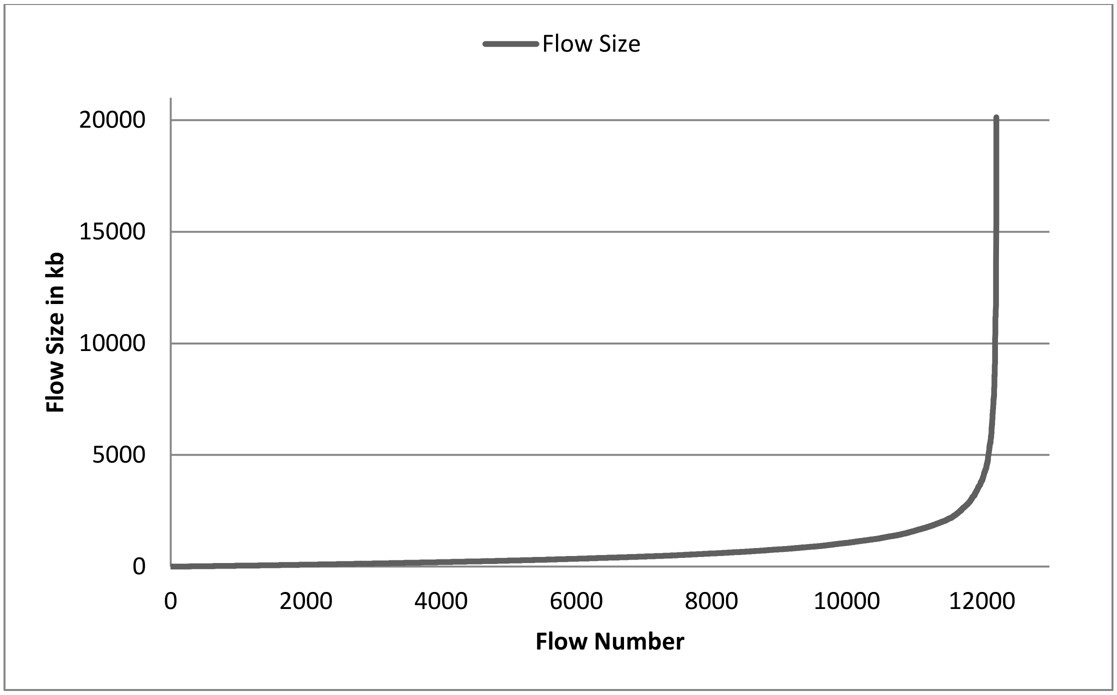


Figure 11: Flow Size Sorted from Smallest to Largest

Network Weatherman Validation

The Network Weatherman is utilized by the system to give predicted values of the future state of the system. The four different network configurations utilized a varying number of Network Weatherman- the smallest number used was 6 and the largest used was 40. Each Network Weatherman needs a MATLAB engine running to support its proper operation during the simulation. The systems that were running the simulation could not handle more than 60 Network Weatherman at a time due to memory constraints.

When setting up the Network Weatherman the Kalman filter has to be tuned to ensure proper operation in the network. The process for tuning the Kalman filter as described in [25] is to iterate through values for X and Y in a 2 by 2 matrix as shown in equation 6.

Equation 3

$$Z = \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}$$

<u>Original Tuned Values</u>

When the simulation was first set up X and Y were evaluated to be 50 and 0.1 respectively. The X and Y values found were for a specific network and traffic profile. When the final network configuration and traffic profile was completed, these values were used. Figure 12 shows three graphs that detail the results for one of the Network Weatherman in the first configuration. The top graph has zoomed in on the first 50 seconds of the simulation run and the next gives a closer look at times 0 to 5 and 10 to 20 seconds, respectively. The Network Weatherman was only able correctly predict the size of the queue about one second prior to it being full. The real problem, as can be seen, is that it incorrectly predicts the value of the queue as the size is shrinking. At 1.7 seconds, it has the queue reaching zero when in fact it does not go below 200.

The other increase and decrease in queue size comes between 20 and 40 seconds. Again, the Network Weatherman does a good job of predicting the full queue, however, when the queue size starts to drop the component falters. The other configurations share similar results with Figure 12.

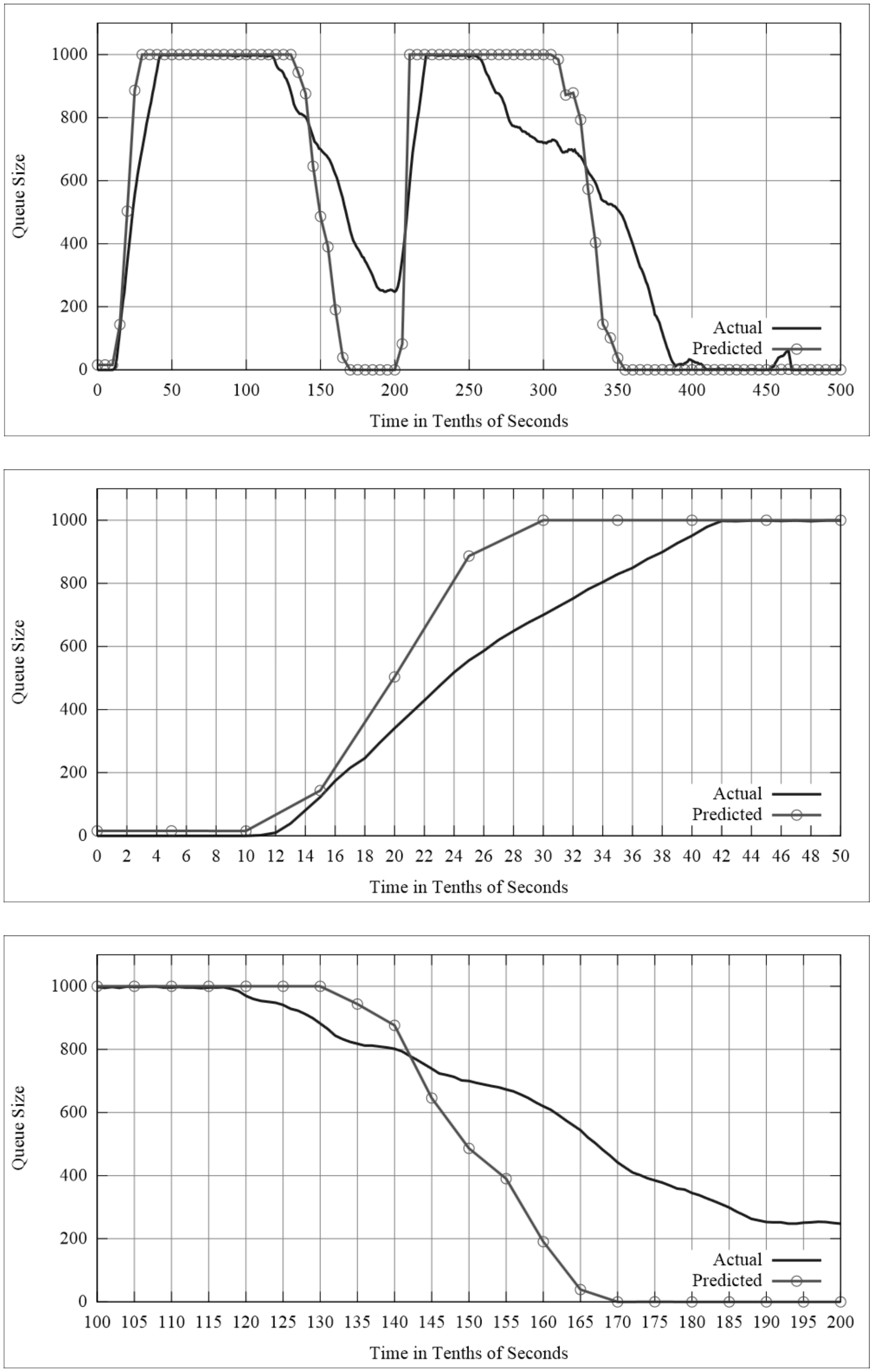


Figure 12: Actual Versus Predicted Queue Size

Retuning of Kalman Filter

During the exploration of the primary simulation results, it was discovered that the original values chosen for the Network Weatherman's Kalman filter were not optimal. The tuning procedure was accomplished again with values ranging from 0.0001 to 1000 for the X and Y variables. The values that work the best for all the queues in the system were found to be X = 85 and Y = 0.001. Figure 13 displays the results for one of the queues in the system. The system was setup to predict 0.6 seconds into the future. When looking at Figure 13 it is important to note that it does not appear that, the Network Weatherman actually provides a predicted value prior to the real value reaching it first.

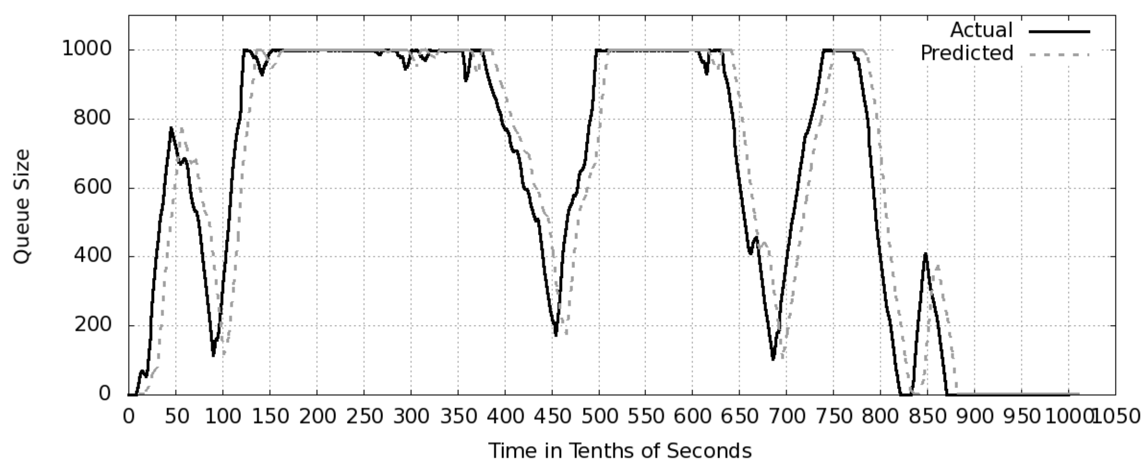


Figure 13: Kalman Filter Retuning Results

This will be discussed in further detail in the secondary research results. The Network Weatherman does a great job of estimating the queue size based on the current queue size for this type of network traffic. The network traffic that was used in [25] was burstier in nature than the network traffic represented in this research. However, the work in [25] does also show an estimate of the queue size, not a prediction.

Primary Simulation Results

This section is broken up into several sections to show the progression of the research. First, results obtained from the four original topologies and the original Network Weatherman tuning. Second, an explanation is given as to why the results were not as expected. Lastly, results are shown again for Topology 2 with modifications made to the simulation configurations.

Topology One

The objective of this research was to determine if the ARAP could improve the overall Quality of Service (QoS) for high priority flows in a network. This section covers the results for the first topology. The flow delays are looked at first followed by a histogram of dropped packets during five-second intervals. Lastly, the total number of packets sent and dropped is provided. The simulation scenario used to generate the figures in this section has a 1:1 ratio for high to low priority flows, the network demand is set to 40 percent and the queue utilization is set to 50 percent.

The results for the average delay for this topology came out as excepted when the NTO was not used and was counter intuitive for when the NTO was used. Figure 14 shows the results when the NTO is utilized and Figure 15 shows the results without the use of NTO nodes. Both show the mean delay for all three runs and contain a 95 percent confidence interval. When NTO is not used, we see a statistically significant difference in the average delay in the NWM Update, as seen in column three of the graph, when compared against No Update and Queue Update. When comparing the results between the NTO and no NTO for the NWM Update it is hard to tell if the NTO caused an improvement with the high priority flows from the graphs by themselves. The numerical, difference for NTO and no NTO as shown in Appendix C Table 5 is 0.16 seconds in favor of the NTO, however, this does not represent an empirically significant difference.

The NWM Update increases the delay felt by the low priority flows by as much as three times. The reason for this threefold increase is due to the extra routes that are now available to the low priority flows. These extra routes will enable some of the paused low priority flows the chance to be restarted. This new route is most likely not as optimal as the first which could lead to a greater delay. The upside to this is that it does lead to more information reaching its destination.

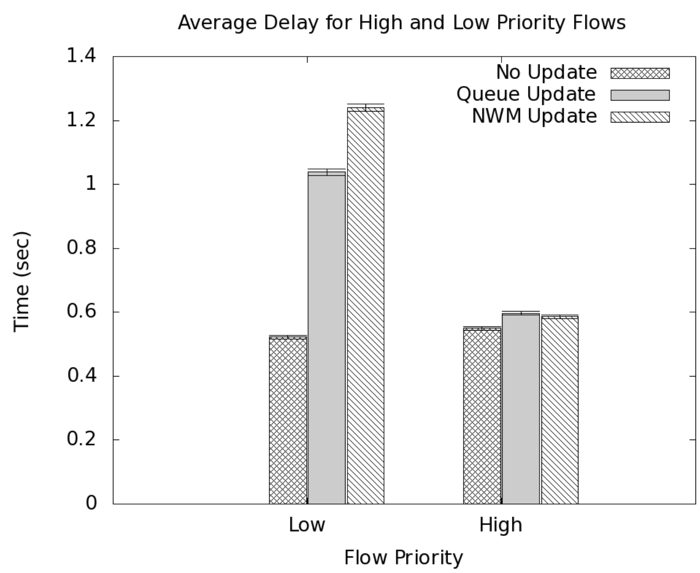


Figure 14: Topology 1 Average Delay with NTO

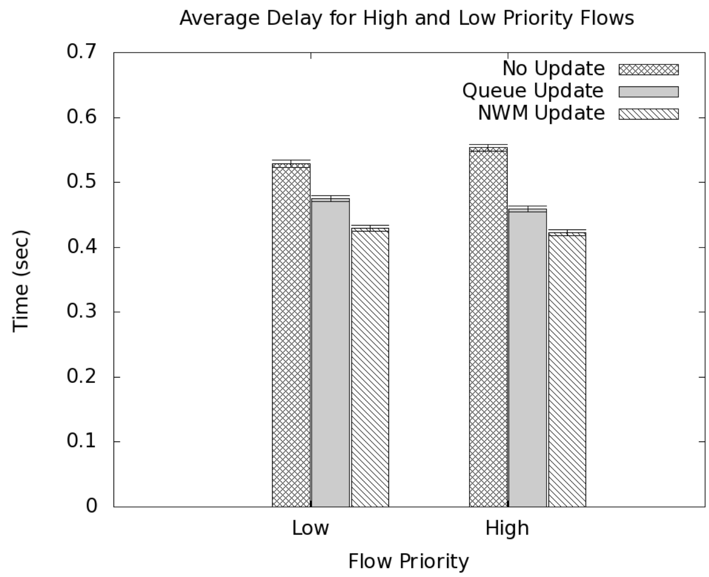


Figure 15: Topology 1 Average Delay without NTO

Figures 16 and 17 show a histogram of dropped packets at five-second intervals for low priority flows. The high priority flows are not shown because 99 percent of the high priority traffic made it to its destination. Figures16 and 17 show the number of dropped packets with and without NTO nodes and edges, respectively. When looking at Figure 16, keep in mind that the NTO nodes and edges are available for use during the following times in the simulation, 20 to 40 seconds and 60 to 90 seconds. During those times, Figure 16 shows that there is a spike in the number of dropped packets experienced by the lower priority flows due to the low priority flows being able to be restarted at those times. When looking at Figure 17, it appears that when no NTO nodes are present there is less packets dropped. This is true, however, a little deceiving in that many of the flows that are dropping packets in Figure 16 are paused and prevented from running in this scenario.

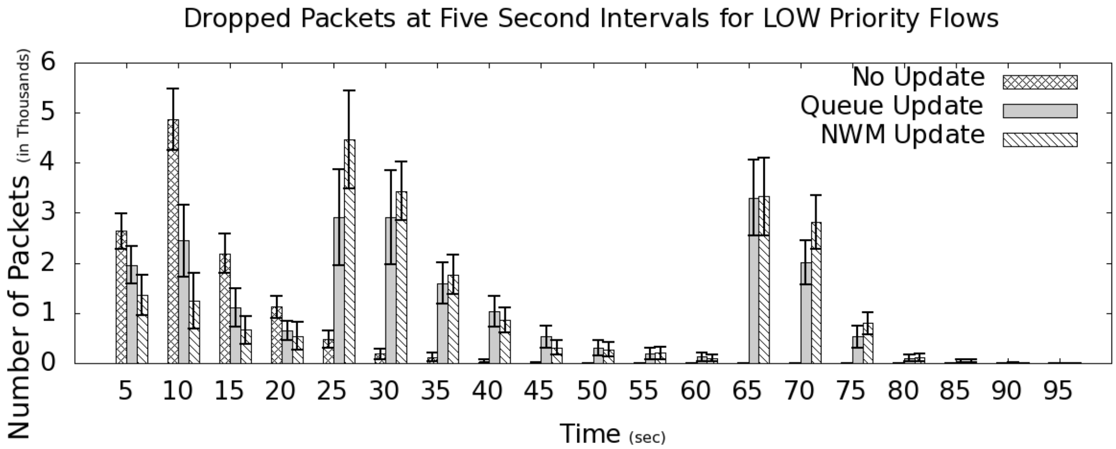


Figure 16: Topology 1 Number of Dropped Packets with NTO

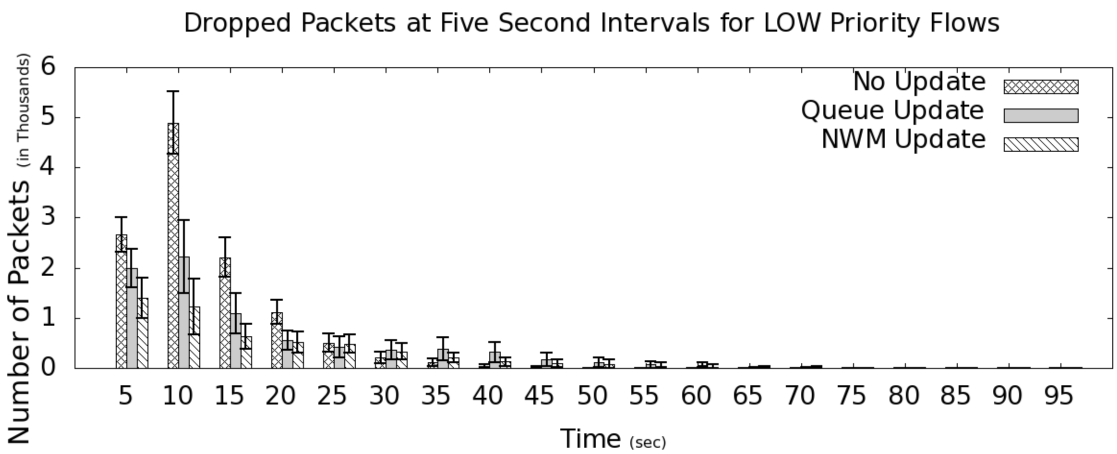


Figure 17: Topology 1 Number of Dropped Packets without NTO

This gives the appearance of degraded performance when the NTO nodes are utilized in the system. However, that depends on what is considered better: either letting nothing get through or allowing some to pass at the prospect that it may be dropped.

Figures 18 and 19 show the results of the number of packets sent versus the number of packets dropped with and without NTO nodes and edges, respectfully.

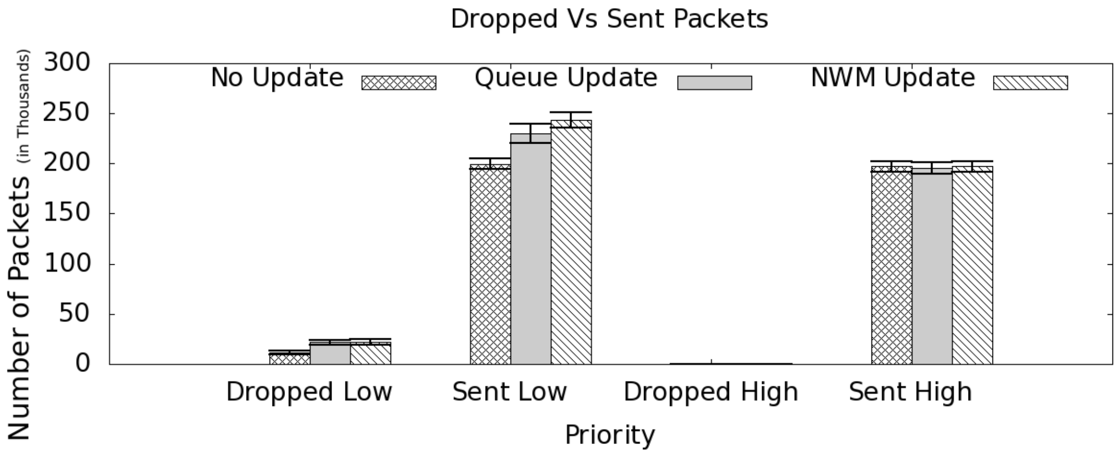


Figure 18: Topology 1 Dropped versus Sent Packets with NTO

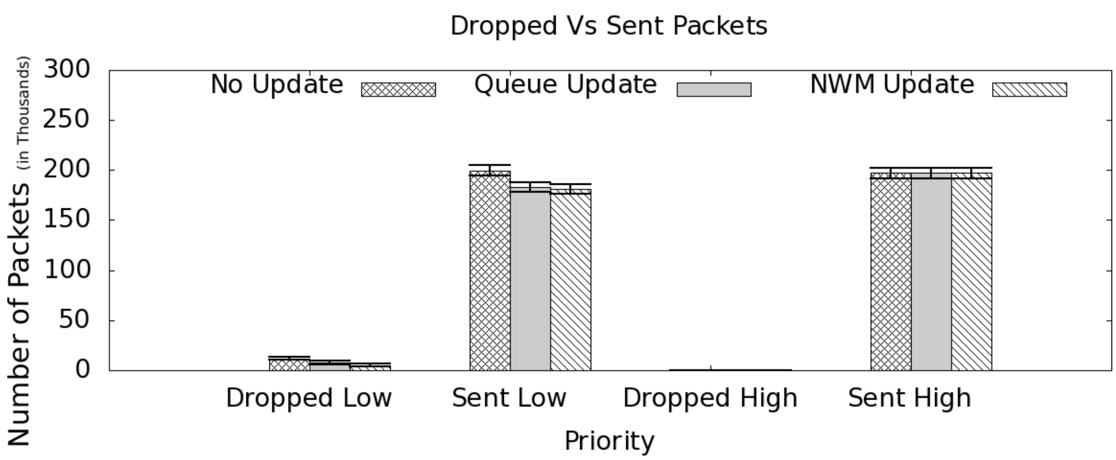


Figure 19: Topology 1 Dropped versus Sent Packets without NTO

Figure 18 shows that it has dropped more overall low priority packets than Figure 19. The numerical value for the drop rate when the NTO nodes are used is 22,362 packets dropped over 243,374 packets sent making the dropped/sent ratio 0.09188. Without the NTO only 5210 packets get dropped over 180,906 packets sent for a dropped/sent ratio of 0.02880. A greater number of packets were sent and made it to their destination; however, the dropped/sent packet ratio did get worse when using the NTO nodes. Only an additional 17,151 packets were dropped, therefore 45,317 more packets reached their destination. The ratio of dropped/sent packets for high priority flows is nearly zero. Table 5 in Appendix C shows the remainder of the results for Topology 1.

Topology Two

The setup of this section is similar to the preceding section. First, the average delay is covered followed by the dropped packets histograms and finally the total number of dropped packets is shown. In Topology 2, and all subsequent topologies, there are at least 20 Kalman filters installed in the network. The simulation scenario used to generate the figures in this section has a 4:1 ratio with a demand set to 65 percent of network bandwidth and queue utilization is set to 70 percent.

The average delay results for Topology 2 are quite different from Topology 1. Figures 20 and 21 show average delay for high and low priority flows with and without NTO nodes, respectively. The two figures show that the NTO and the NWM make no difference in this new topology with respect to high priority flows. The numerical data provides a similar picture for delay in that the difference in the values for high priority flows with the above-mentioned setup is an improvement of 0.04785 seconds and the low priority flows improvement is 0.0559 seconds when utilizing the NTO nodes. When looking at the comparison between No Update and the NWM update the improvement is only 0.008 seconds when using the NTO and 0.01024 without. The low priority flows show a similar magnitude but an opposite effect is present which it is to be expected because they are taking the optimal routes. The numerical data for these charts can be found in the Appendix in Table 6.

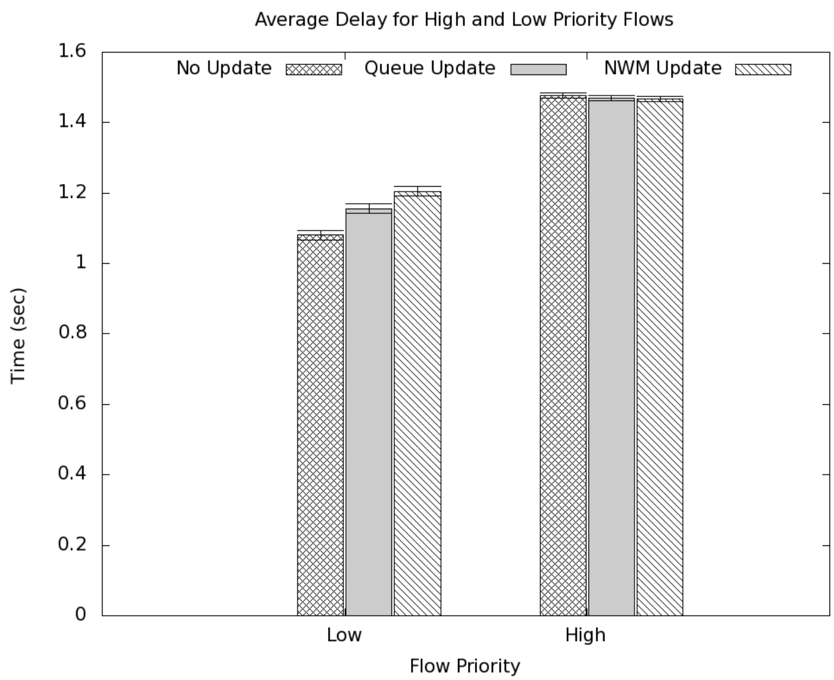


Figure 20: Topology 2 Average Delay with NTO

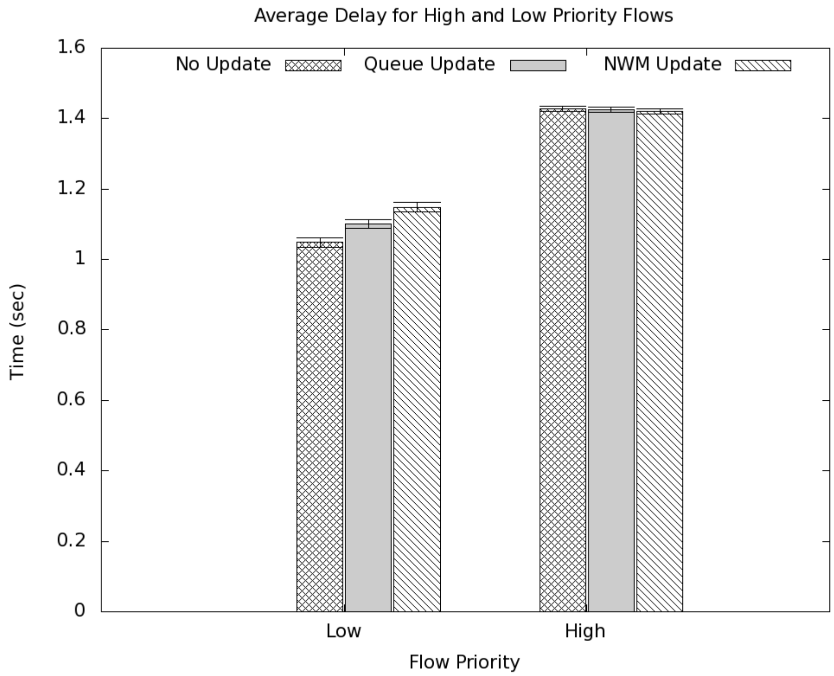


Figure 21: Topology 2 Average Delay without NTO

High priority packet loss is shown in Figures 22 and 23 that show a histogram of dropped packets at five-second intervals with the same scenario as Figures 20 and 21. Figure 22 represents the number of packets dropped when utilizing the NTO and Figure 23 represents the number of packets dropped when not using the NTO.

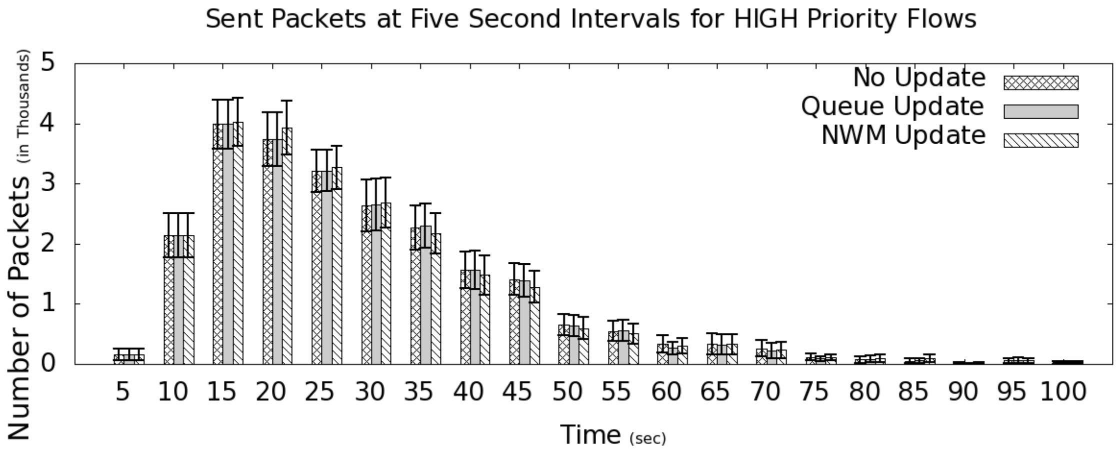


Figure 22: Topology 2 High Priority Packet Loss with NTO

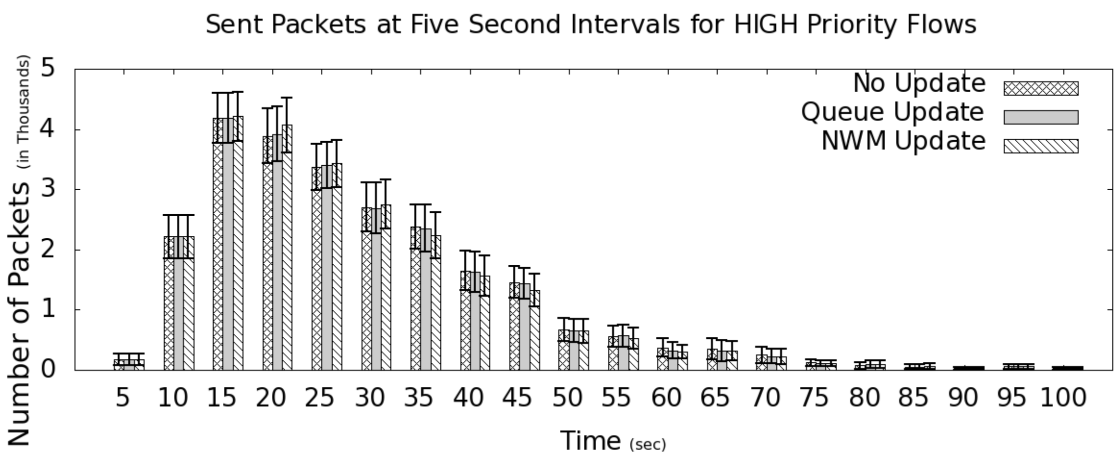


Figure 23: Topology 2 High Priority Packet Loss without NTO

Figures 22 and 23 again look identical, however, there is a slight improvement in the number of packets dropped during the majority of the five second intervals when using the NTO. The low priority flows also behave in a similar manner to the high priority flows however, they drop more packets.

The dropped versus sent packets for high and low priority flows are shown in Figures 24 and 25 for this scenario. Figure 24 is without the NTO and Figure 25 is with the NTO. Figures are again practically identical, however, there is a slight improvement in the amount of packets that reached their destination.

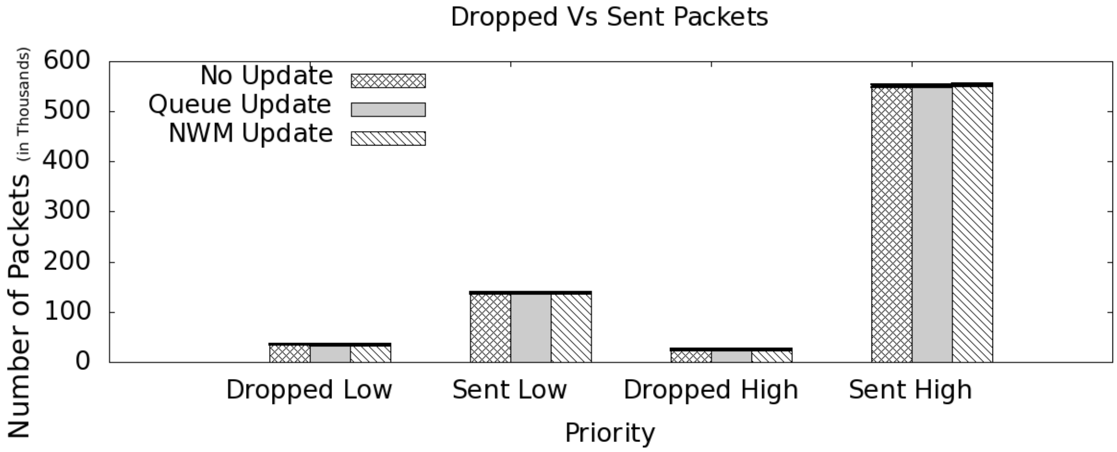


Figure 24: Topology 2 Dropped versus Sent Packets without NTO

There are improvements in the number in the total number of dropped packets for both low and high priority flows. However, those improvements lack any statistical significance and they only make up 9.6 percent of the low priority dropped traffic and 3.7 percent of the high priority dropped traffic.

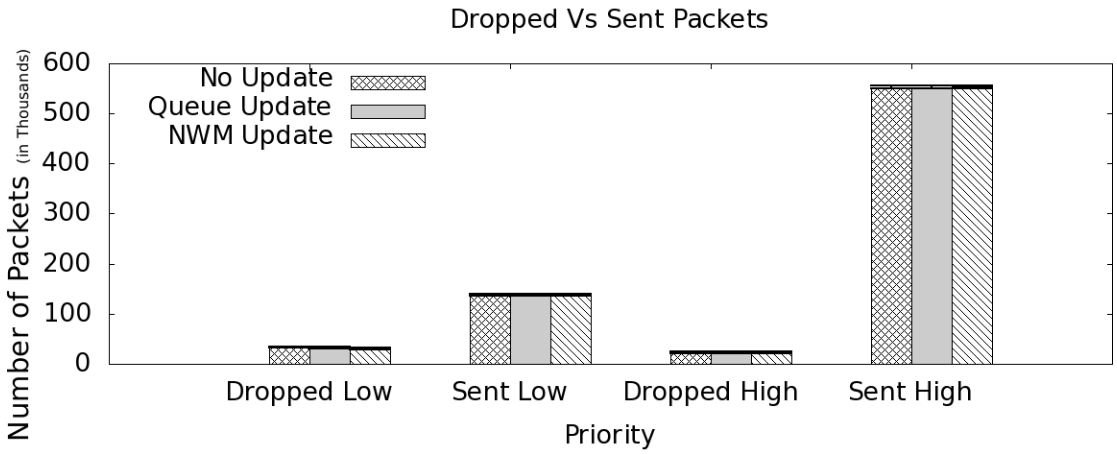


Figure 25: Topology 2 Dropped versus Sent Packets with NTO

Summary of Results for Topology 3 and Topology 4

Topology 3 and topology 4 results strongly resemble those of Topology 2. The numerical results for Topology 3 and 4 can be found in Appendix C in Table 7 and 8 respectively.

The lack of any statistically significant difference in topologies two through four led to the question of why is there no statistical difference between No Update, Queue Update and NWM Update. One would expect to see that the Queue Update and the NWM Update might not be that different due, in part, to the fact that the predictions are only valid approximately 0.6 seconds into the future. The fact that two potential improvements appeared on the surface to do no better than nothing at all led to the next couple of sections where some additional analysis was completed.

Additional Analysis

With Topologies 2 through 4 showing no statistically significant results and Topology 1 only showing significant results in the delay area. Some additional analysis was completed to help answer why the Queue Update and NWM Update could not do any better than nothing at all.

Edge Betweenness

Something that all the graphs had in common was that they all had one edge linking a small group of nodes to a central group of nodes. The idea is that maybe the single edge is where all the packets are being dropped and is related to edge betweenness. Where edge betweenness is a measure of how much a particular edge is needed by source node to reach a destination node on the other side. The higher the edge betweenness value the more important that edge is to the connectivity of the graph. The edge betweenness values for Topology 2 range from 0 to 200. Another way of looking at it in this case is that an edge receiving a betweenness value of 200 is an edge that is highly utilized by Topology 2. If an edge has a high betweenness value than there is not many paths around that edge. This can cause some issues with the ARAP routing of high priority flows. The multicommodity routing algorithm attempts to spread the flows out over the network, however, the spreading out of flows is hindered by edges with high betweenness values.

The scenario used to look at edge betweenness has a 4:1 ratio, demand is set to 65 percent and queue utilization is 70 percent because it is the best example of what I was looking for in my results. The other scenarios do not give a clear picture of this especially when the ratio is 1:1. As can be seen from Figure 26 the edges with a betweenness value greater than 100 drop considerably more packets than those with 100 or less. The same can be said about Figure 27, which shows the same run, but without the NTO.

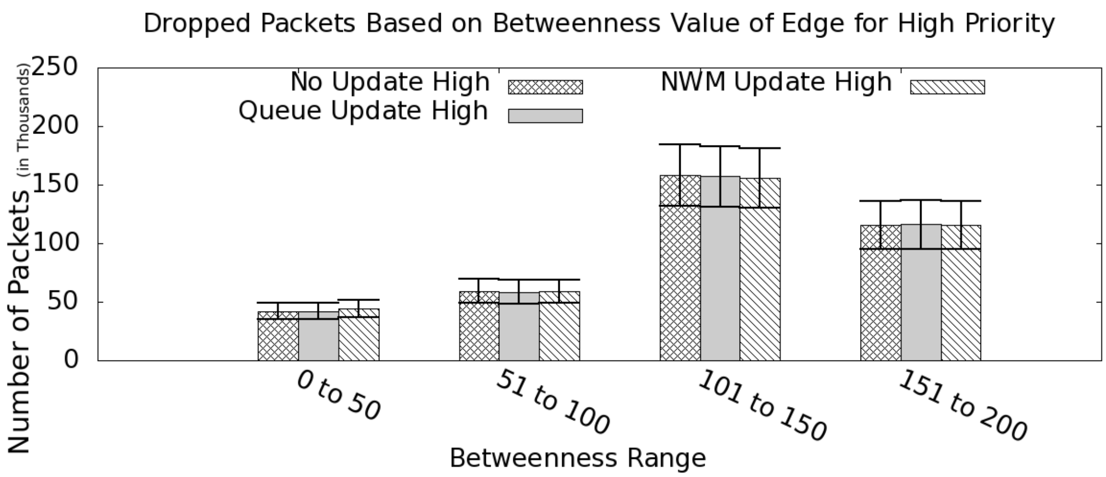


Figure 26: Topology 2 Dropped Packets Based on Edge Betweenness Value with NTO

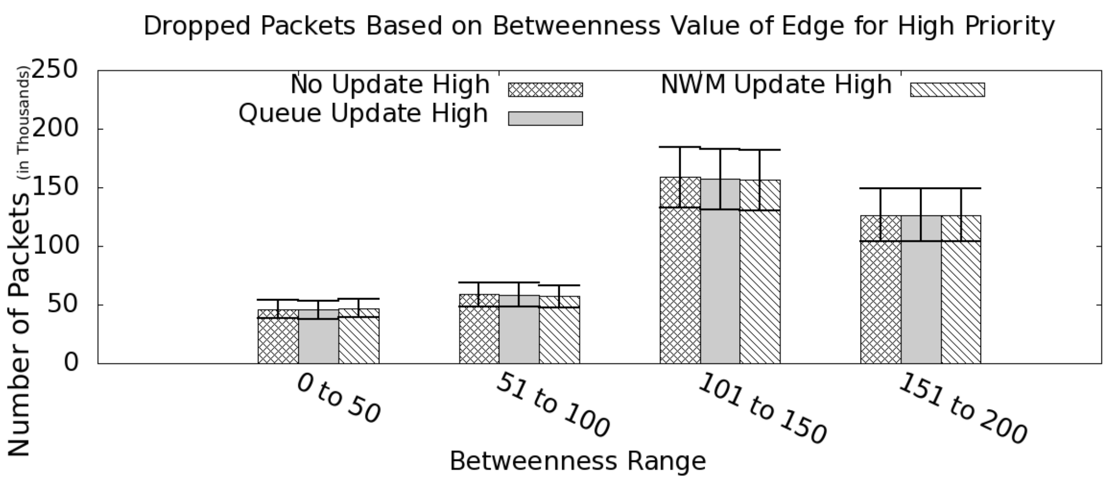


Figure 27: Topology 2 Dropped Packets Based on Edge Betweenness Value without NTO

Conclusions from Additional Analysis

Another possible answer discovered when looking at the betweenness values was that when the edges with NWM on them reached their threshold low priority flows were routed around the congested edges. The significance of this is that in Topologies 2 through 4, many of the edges that picked up the additional load did not have a NWM on them and when queues on those edges reached capacity they were unable to tell the ARAP system that they were overloaded and thus continued to drop packets without any relief. Lastly, the configuration setup was reevaluated from Chapter III and it was found that priority droptail queues were used for all simulation configurations on each edge of the network and the multicommodity flow algorithm was used to route high priority flows in all three configurations.

With exception of Topology 1 the results from the primary simulation runs suggested that there is no difference between the results obtained for the three simulation configurations No Update, Queue Update and NWM Update. With this knowledge and the reevaluated information, it is concluded that the ARAP system was being compared to variations of itself and not a normal network for the following reasons. First, the No Update configuration is also utilizing the Fleischer algorithm to spread out the high priority flows more evenly in the network. A normal network does not utilize an algorithm like Fleischer's, networks use shortest path no matter the flow type. Second, the use of priority queues on all the edges in every configuration could potentially mask the effects of the ARAP.

Secondary Simulation Results

To correct for the invalid assumptions made during the setup of the primary results all the simulations for Topology 2 is rerun with No Update, utilizing only the shortest path to route flows in the network and no priority queues is utilized. For the Queue Update, the Fleischer algorithm is used as well as priority queues but only on the edges that have the ability to send the ARAP an updated queue size value. The NWM

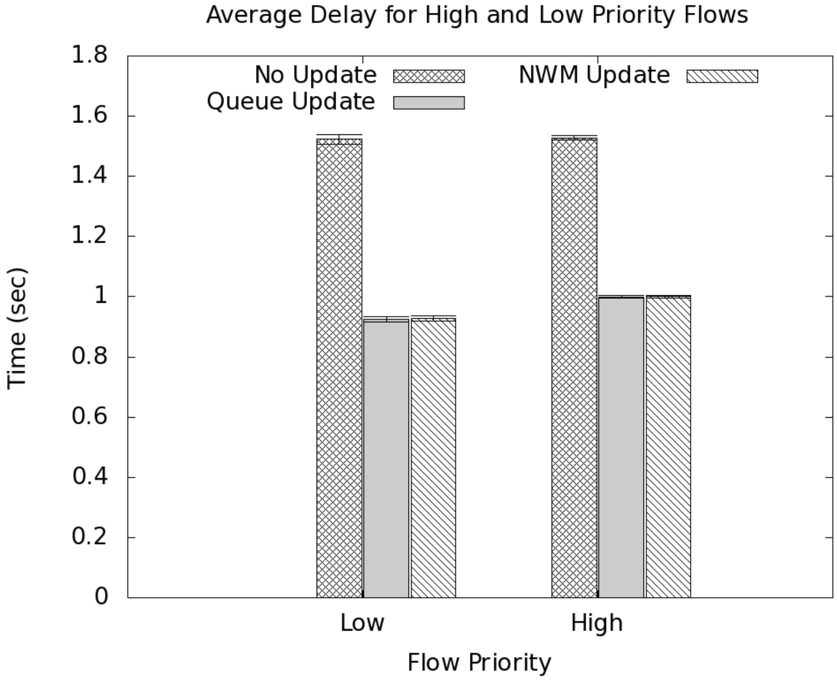
Update, also uses the Fleischer algorithm and priority queues are used only on the edges that also contained the NWM.

Secondary Topology Two

This section covers the results for when Topology 2 was run again with priority queues removed expect for edges that have the ability to send the ARAP a queue size update. The simulation setup for all the figures of this section of these charts is ratio of 4:1 with a demand set to 65 percent of the network bandwidth and the queue utilization set to 70 percent. Each figure will contain both with and without the NTO information on the top and bottom, respectively.

In Figure 28, the delay shows that the NWM does in fact make a significant difference in the delay that the low and high priority flows experience. The graphs contain a 95 percent confidence interval. The top graph in Figure 28 displays the results when the NTO is used and the lower graph in the figure displays the results when no NTO is used. It is hard to see from the graphs in Figure 28 but the NTO also improves the delay by 0.05 seconds. Figure 29 more clearly shows the improvement that is provided by using the NTO in conjunction with the NWM. Figure 29 also contains error bars with a 95 percent confidence interval and shows that the high priority flows do experience a higher delay than the lower priority flows that are attributed to the spreading out of the high priority flows in the network.

Figure 29 also shows that using predicted or real-time values have no significant impact on the delay felt by the flows in the network. In Chapter V, an explanation is given to explain the lack of significance with respect to real-time versus predicted.



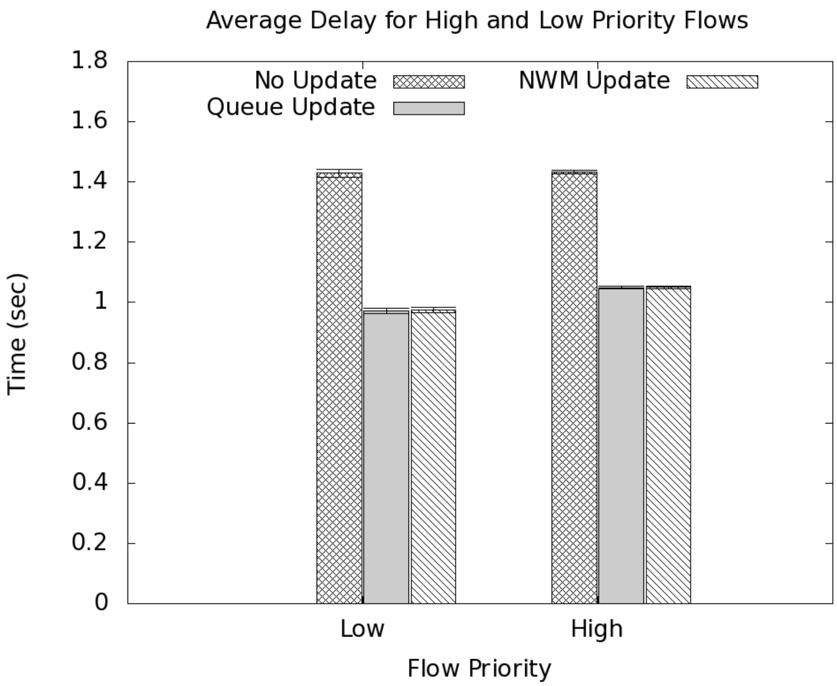


Figure 28: Secondary Average Delay Results for Topology 2 with and without NTO

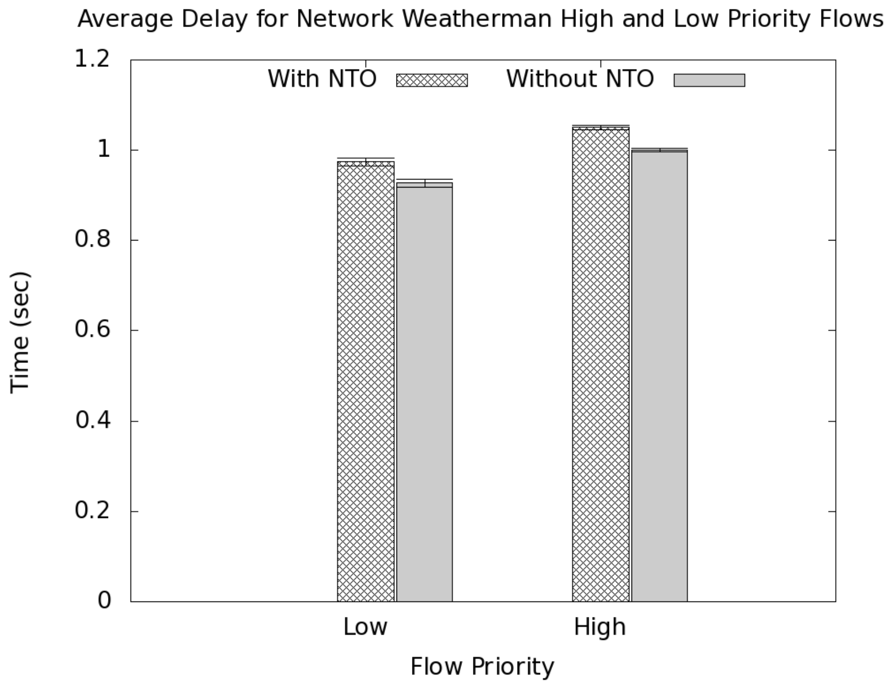


Figure 29 Average Delay With and Without NTO for Network Weatherman

When looking at the dropped packets for the five-second interval histogram for high priority flows there is a significant improvement shown between the NWM and the No Update configuration as seen in the top of Figure 30. The low priority flows also experience a significant decrease in the number of dropped packets over the No Update configuration. The error bars in Figure 30 represents a 90 percent confidence interval.

Figure 30 also shows that using the real-time versus predicted values to change the routes of the lower priority flows contain no significant difference. An explanation for this is covered in Chapter V.

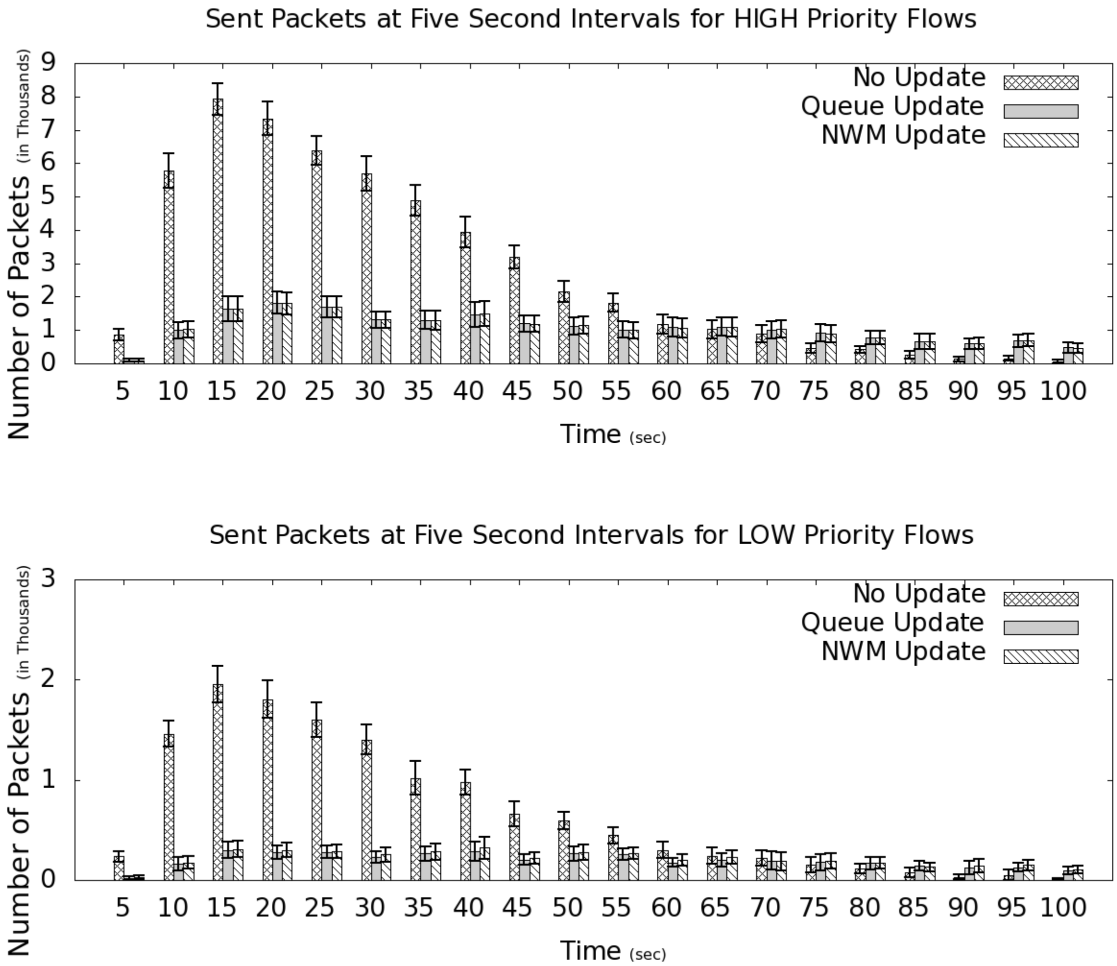
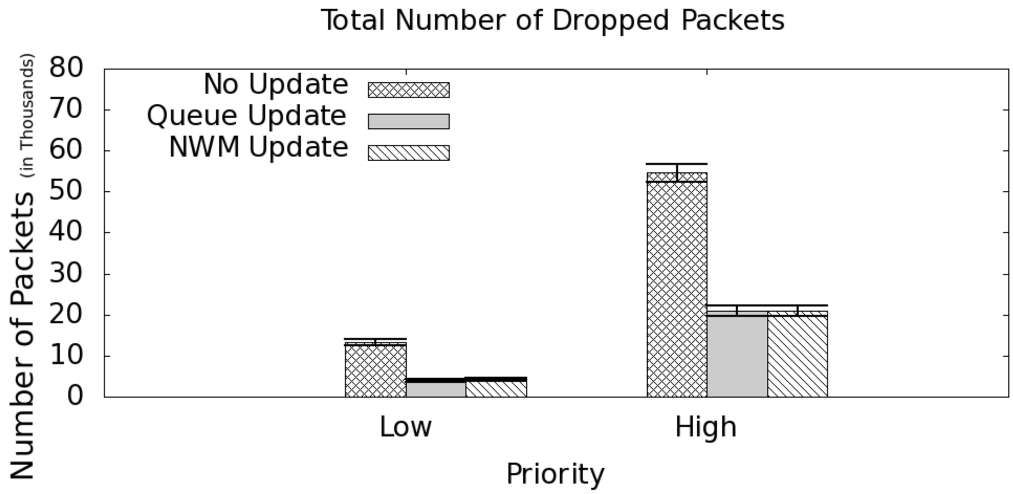


Figure 30: Secondary Results for Topology 2 Dropped Packets Histogram for Low and High Priority Flows

The total number of packets dropped and sent for this configuration is displayed in Figure 31. The error bars in Figure 31 represent a 95 percent confidence interval. The top graph in Figure 31 is the number of dropped packets with the NTO in use and the bottom graph is without the NTO.



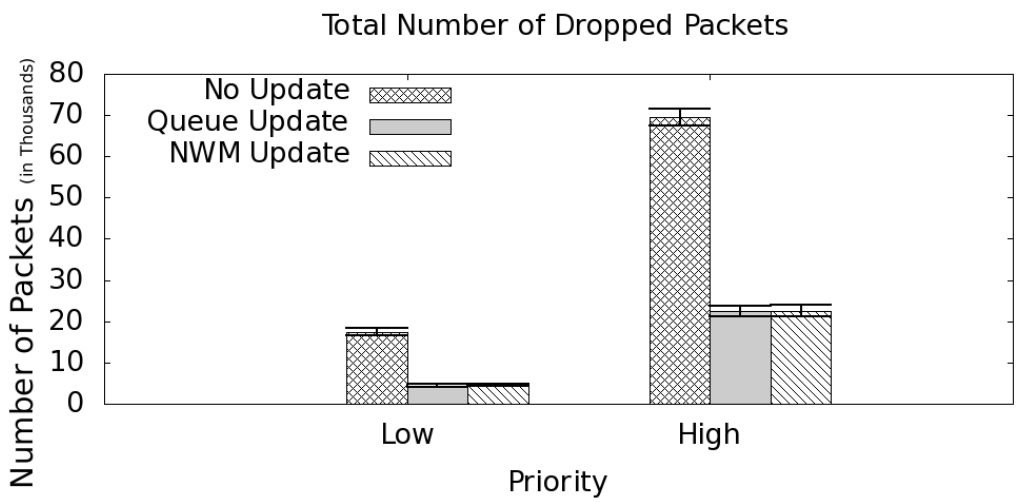


Figure 31: Secondary Results Topology 2 Dropped versus Sent Packets

Summary

In summary, the beginning of this chapter highlighted three different simulation configurations that showed little to no difference when using the ARAP system. Upon further investigation, it was found that these results appeared irrelevant because all three configurations contained the routing algorithm of the ARAP system and the priority drop tail queues on all the edges masked the impact of the ARAP system. In the primary results section we were able to determine that the NTO does improve the amount of data that is delivered to its destination but the results are not always statically significant. The NWM aspect of the ARAP system displayed no distinct difference between the other to simulation configurations in the primary results section. This is due to the utilization of the multicommodity flow algorithm in all three initial network configurations.

The secondary results section shows that the ARAP system does provide a statistically significant difference between it and the No Update configuration. Whether the ARAP system uses predicted or real-time queue size updates appears to have no effect on the results.

In the validation section, it was shown that the NWM was able to accurately estimate the queue sizes. However, the NWM was unable to provide predicted queue sizes for this network setup.

V. Conclusions and Recommendations

Chapter Overview

This chapter discusses the conclusions of the research presented in the previous chapters. Some new ideas and potential follow on tasks have been discovered along the way and are covered in more detail in the future research section of this chapter.

Conclusions of Research

In this thesis, an Adaptive Routing Algorithm for Priority (ARAP) flows in a Network was presented as a way to improve the quality of service in the realm of flow delays and packet loss rates. In order to accomplish this, three previous ideas including the Network Tasking Order (NTO), Network Weatherman (NWM) and multicommodity flow algorithm were put together to create a routing agent that utilized the information from those products to direct information flows in the network around congested edges to less congested edges when possible. If redirection of the information flows were not possible, the lower priority flows were stopped to allow the higher priority flows better access to the network.

The first objective of this research was to develop a priority aware routing protocol for network flows. This objective was accomplished by creating a flow agent called the ARAP that was able to utilize the information provided the NTO and NWM. Chapter III outlined and discussed the setup of the simulation and components used including the ARAP. The NTO provides the information required to categorize each flow into the appropriate priority and gives source destination pairs for high priority flows. The NWM provides predicted queue sizes that enabled the rerouting of flows around potentially congested edges. This simulation provides a foundation for future work in the area of network optimization.

The second objective of this research was to improve the quality of service for higher priority flows in the network. Chapter IV provided the simulation results and was broken down into two results sections. The primary results section compared three different variations of the ARAP system. The results from the primary section did not provide the insight that was intended for this research initially. A mistake was made by comparing three different variations of the ARAP system however, some valuable information was obtained from the data. The data from the primary results section shows that the individual components of the ARAP system potentially contribute varying amounts of improvement. I suggest that the multicommodity flow algorithm adds the most value to the system but this is left to be proved during another research effort.

The secondary results section shows that the ARAP system does in fact provide better quality of service to the higher priority flows in the network. However, there is no difference between using the predicted queue size from the NWM or real-time queue sizes provided by the queues.

The third objective of this research was to integrate the prediction of queue sizes into the routing protocol. The integration of the NWM was accomplished by integrating the MATLAB 2007 libraries and the MATLAB 2010 engine. Chapter IV covered the results of this integration. The primary validation did not show proper operation of the Kalman filter, however, the secondary validation was able to show that the Kalman filter was correctly integrated into the system and was able to give accurate estimates of the queue sizes. However, the Network Weatherman was unable to predict these estimates into the future for this style network and traffic pattern. This explains why there was no difference between the results for the NWM update and Queue update network configurations. The Kalman filter does a great job of estimating the size of the queues when properly tuned, however, this research has found that the NWM does not provide predicted values for the network queues in this research.

Recommendations for Future Research

This research primary focus was on high and low priority levels. It would be difficult to assign all the information flowing across the military network into only two priority levels therefore further research on this subject would be to expand on the number of priority levels the system could handle.

Further research needs to be done in the area of the NWM to determine its ability to apply it to network applications on this scale. The work done in [25] used traffic that caused the queue sizes to fluctuate in a much greater rate than those in this research. It could be that the NWM cannot accurately predict this style of traffic flow where the queues are not changing at significant rates. If this were the case, it would also prove useful to automate the tuning of the NWM so that if can become a self-correcting.

The results of this research suggest that the value added by each component of the ARAP system varies. Therefore, it would be relevant to look at the value added by each component singularly and in combinations to see if the system can be paired down to less than the three components.

Appendix A

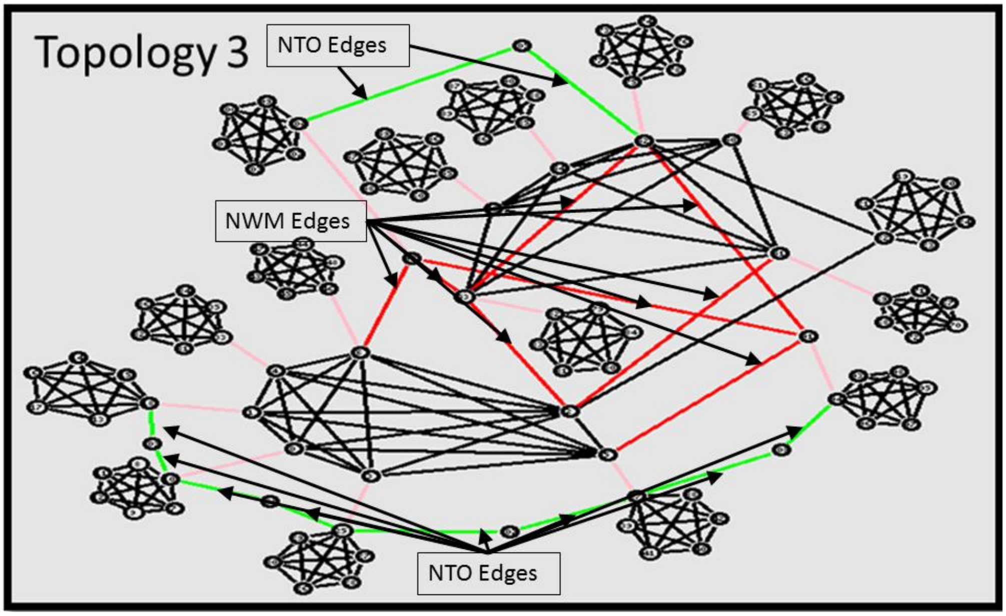


Figure 32: Topology 3

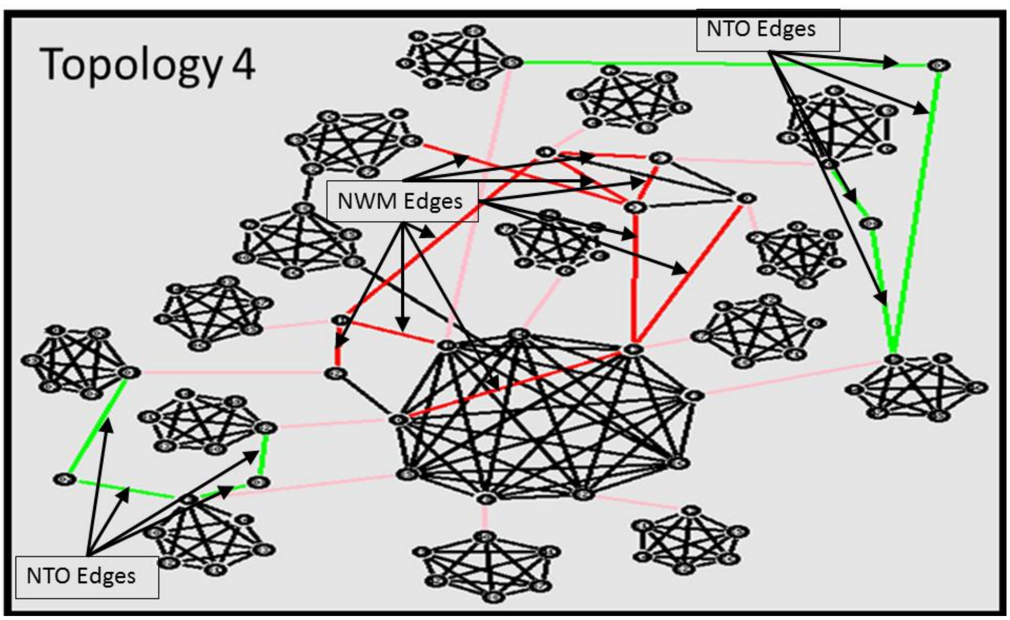


Figure 33: Topology 4

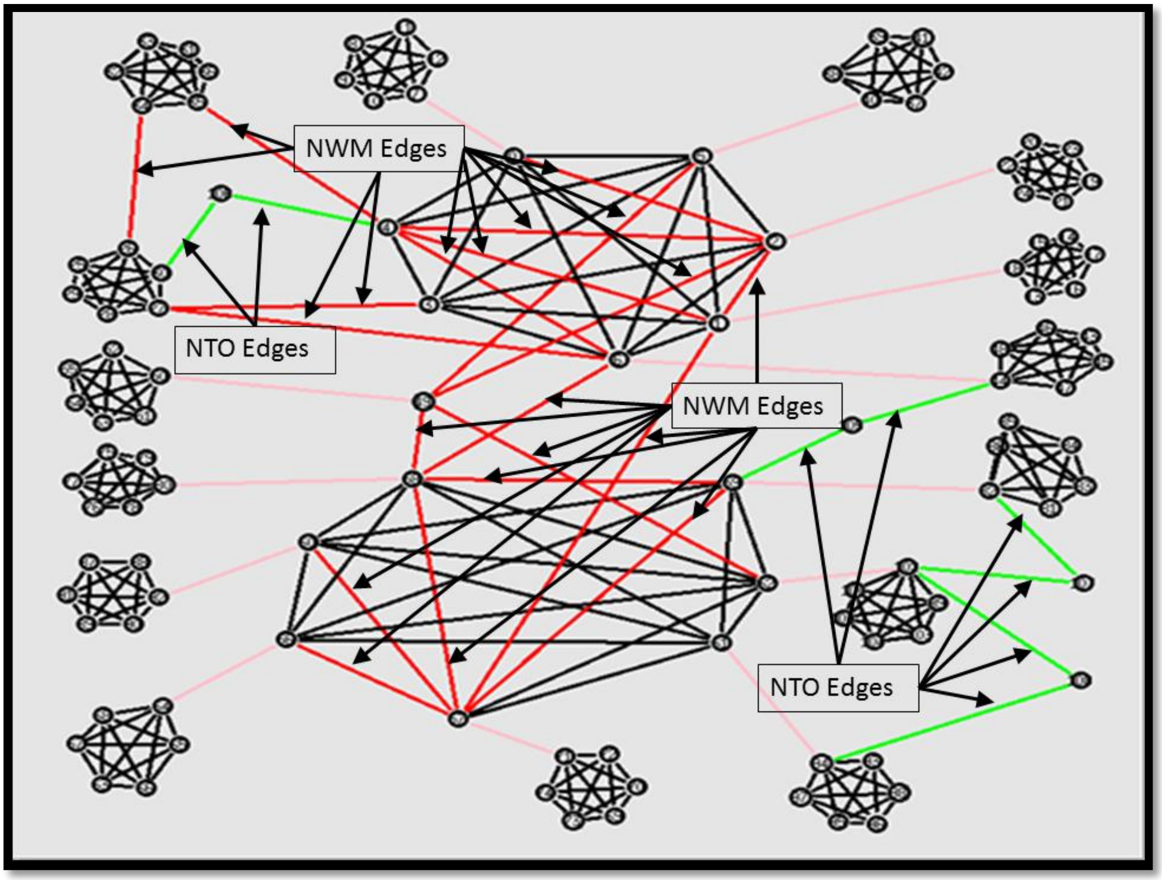


Figure 34: Topology 2 with Additional NWM

Appendix B

Script used to Generate Topology 2 and Topology 4

```
# <method keyword><number of graphs> [<initial seed>]
# <# stubs/trans node><#rand. t-s edges><#rand. s-s edges>
# <n><scale><edgemethod><alpha> [<beta>] [<gamma>]
#
ts 10 52
1 0 2
3 10 4 0.5 0 0
5 10 4 0.4 0 0
6 10 4 0.4 0 0
```

Output of the GA Tech script which generated the Topologies.

Topology 4

```
# Generated by sgb2ns, created by Polly Huang
# GRAPH (#nodes #edges id uuvvww xx yyzz):
# 105 582
transtub(0,1,0,2,{3,92,4,0.500,0.000,0.000},{5,46,4,0.400,0.000,0.000},{6,46,4,0.400,0.0
00,0.000}) 92 1 1 1
```

Topology 2

```
# Generated by sgb2ns, created by Polly Huang
# GRAPH (#nodes #edges id uuvvww xx yyzz):
# 105 580
transtub(0,1,0,2,{3,92,4,0.500,0.000,0.000},{5,46,4,0.400,0.000,0.000},{6,46,4,0.400,0.0
00,0.000}) 92 1 1 1
```

Appendix C

Table 5: Topology 1 Results

High Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00002	0.55342	0.00005	0.45934	0.00003	0.42227
Demand 0.4 Utility 0.7	0.00001	0.55219	0.00001	0.4765	0.00003	0.4416
Demand 0.65 Utility 0.5	0.00584	1.36032	0.00777	1.28951	0.00795	1.29506
Demand 0.65 Utility 0.7	0.00616	1.36184	0.00753	1.29132	0.00776	1.29474
Ratio 4:1						
Demand 0.4 Utility 0.5	0.00796	0.56785	0.00847	0.52417	0.00843	0.51691
Demand 0.4 Utility 0.7	0.008	0.56529	0.00839	0.53016	0.0085	0.52051
Demand 0.65 Utility 0.5	0.08918	1.44124	0.08924	1.40353	0.08924	1.40888
Demand 0.65 Utility 0.7	0.08875	1.44205	0.0891	1.40812	0.08926	1.40719
High Priority Flows						
with NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00001	0.54934	0.00004	0.59687	0.00003	0.58612
Demand 0.4 Utility 0.7	0.00001	0.54719	0.00007	0.5987	0.00004	0.58888
Demand 0.65 Utility 0.5	0.00629	1.35646	0.00785	1.55427	0.00789	1.54995
Demand 0.65 Utility 0.7	0.00603	1.35577	0.00779	1.55659	0.00801	1.5676
Ratio 4:1						
Demand 0.4 Utility 0.5	0.00806	0.56581	0.00836	0.57773	0.00852	0.56691
Demand 0.4 Utility 0.7	0.00782	0.56478	0.00853	0.57654	0.00845	0.56474
Demand 0.65 Utility 0.5	0.08887	1.43911	0.08783	1.57738	0.08743	1.57873
Demand 0.65 Utility 0.7	0.08889	1.43953	0.08782	1.57252	0.08818	1.57914
Low Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.05899	0.52853	0.04261	0.47482	0.0288	0.42899
Demand 0.4 Utility 0.7	0.0587	0.52806	0.04604	0.4835	0.03876	0.46323
Demand 0.65 Utility 0.5	0.31178	1.121	0.24158	1.05063	0.24769	1.08257
Demand 0.65 Utility 0.7	0.3116	1.12153	0.24364	1.05625	0.24761	1.07512
Ratio 4:1						
Demand 0.4 Utility 0.5	0.1171	0.43388	0.08319	0.39825	0.07583	0.39732
Demand 0.4 Utility 0.7	0.11836	0.42958	0.09295	0.4032	0.08005	0.39646
Demand 0.65 Utility 0.5	0.44504	0.68373	0.39287	0.66787	0.39359	0.68379
Demand 0.65 Utility 0.7	0.44674	0.68399	0.39643	0.67241	0.39587	0.68497
Low Priority Flows						
With NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.05839	0.52186	0.09464	1.03841	0.09188	1.24005
Demand 0.4 Utility 0.7	0.05811	0.52094	0.09579	1.0296	0.08882	1.23216
Demand 0.65 Utility 0.5	0.30692	1.10175	0.34941	2.49817	0.35079	2.66491
Demand 0.65 Utility 0.7	0.30664	1.10168	0.34154	2.38349	0.3494	2.51709
Ratio 4:1						
Demand 0.4 Utility 0.5	0.11711	0.43073	0.08587	0.54856	0.0816	0.55478
Demand 0.4 Utility 0.7	0.11794	0.43052	0.08963	0.54713	0.07428	0.52987
Demand 0.65 Utility 0.5	0.44175	0.6725	0.35471	1.41594	0.35191	1.41307
Demand 0.65 Utility 0.7	0.44057	0.67562	0.35231	1.35416	0.35484	1.35777

Table 6: Topology 2 Results

High Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.0007	0.47195	0.0007	0.47419	0.00075	0.4861
Demand 0.4 Utility 0.7	0.0007	0.47195	0.0007	0.47155	0.0008	0.47807
Demand 0.65 Utility 0.5	0.0068	1.47095	0.00657	1.45161	0.00687	1.4392
Demand 0.65 Utility 0.7	0.0068	1.47095	0.00669	1.45565	0.00686	1.44621
Ratio 4:1						
Demand 0.4 Utility 0.5	0.00262	0.44483	0.00259	0.44472	0.00263	0.44551
Demand 0.4 Utility 0.7	0.00262	0.44483	0.00258	0.44477	0.00263	0.44593
Demand 0.65 Utility 0.5	0.0446	1.47747	0.04411	1.47443	0.0443	1.46699
Demand 0.65 Utility 0.7	0.0446	1.47747	0.04444	1.47025	0.04411	1.46723
High Priority Flows						
with NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00069	0.4626	0.00069	0.46506	0.00073	0.47261
Demand 0.4 Utility 0.7	0.00069	0.4626	0.00069	0.46273	0.00072	0.46746
Demand 0.65 Utility 0.5	0.00658	1.43051	0.00637	1.41633	0.00638	1.39913
Demand 0.65 Utility 0.7	0.00658	1.43051	0.00647	1.42648	0.00665	1.40727
Ratio 4:1						
Demand 0.4 Utility 0.5	0.00252	0.43495	0.00251	0.43493	0.00252	0.43282
Demand 0.4 Utility 0.7	0.00252	0.43495	0.00248	0.43503	0.00251	0.43453
Demand 0.65 Utility 0.5	0.04275	1.42785	0.04259	1.42547	0.04249	1.41986
Demand 0.65 Utility 0.7	0.04275	1.42785	0.0425	1.42473	0.04252	1.41965
Low Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01469	0.49192	0.01859	0.50993	0.02578	0.58298
Demand 0.4 Utility 0.7	0.01469	0.49192	0.0155	0.49533	0.02101	0.54376
Demand 0.65 Utility 0.5	0.17761	1.38878	0.20191	1.53542	0.20298	1.54343
Demand 0.65 Utility 0.7	0.17761	1.38878	0.19899	1.49952	0.20624	1.55666
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01962	0.43799	0.01916	0.43719	0.02483	0.47117
Demand 0.4 Utility 0.7	0.01962	0.43799	0.01928	0.43779	0.02628	0.47416
Demand 0.65 Utility 0.5	0.25564	1.08036	0.25812	1.2147	0.24071	1.18848
Demand 0.65 Utility 0.7	0.25564	1.08036	0.256	1.15604	0.24622	1.20478
Low Priority Flows						
With NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01389	0.48415	0.01727	0.50362	0.02307	0.55343
Demand 0.4 Utility 0.7	0.01389	0.48415	0.01479	0.48808	0.01933	0.52585
Demand 0.65 Utility 0.5	0.17148	1.35748	0.19007	1.48136	0.19913	1.5203
Demand 0.65 Utility 0.7	0.17148	1.35748	0.19264	1.4842	0.19371	1.51688
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01808	0.42862	0.01801	0.42842	0.02085	0.44016
Demand 0.4 Utility 0.7	0.01808	0.42862	0.01774	0.4288	0.0232	0.45451
Demand 0.65 Utility 0.5	0.24469	1.04833	0.23887	1.13374	0.22421	1.15727
Demand 0.65 Utility 0.7	0.24469	1.04833	0.24264	1.10154	0.22453	1.14888

Table 7: Topology 3 Results

	High Priority Flows					
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00296	0.98816	0.00298	0.98615	0.00289	0.97283
Demand 0.4 Utility 0.7	0.00296	0.98816	0.00296	0.98823	0.00294	0.97672
Demand 0.65 Utility 0.5	0.02516	2.51012	0.02527	2.49508	0.02544	2.4662
Demand 0.65 Utility 0.7	0.02516	2.51012	0.02519	2.50248	0.02531	2.46876
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01392	0.93989	0.01389	0.93823	0.01391	0.93189
Demand 0.4 Utility 0.7	0.01392	0.93989	0.01392	0.93989	0.01393	0.93418
Demand 0.65 Utility 0.5	0.09739	2.53102	0.09748	2.52075	0.09695	2.5063
Demand 0.65 Utility 0.7	0.09739	2.53102	0.09739	2.52144	0.09694	2.50483
	High Priority Flows					
with NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00283	0.95664	0.00282	0.95528	0.0028	0.94258
Demand 0.4 Utility 0.7	0.00283	0.95664	0.00282	0.95625	0.00283	0.94504
Demand 0.65 Utility 0.5	0.02297	2.42602	0.02303	2.4173	0.02314	2.38524
Demand 0.65 Utility 0.7	0.02297	2.42602	0.02306	2.42822	0.02317	2.38557
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01294	0.90112	0.01294	0.89957	0.01292	0.89261
Demand 0.4 Utility 0.7	0.01294	0.90112	0.01294	0.9013	0.01294	0.89366
Demand 0.65 Utility 0.5	0.08913	2.43943	0.08921	2.43845	0.0886	2.4201
Demand 0.65 Utility 0.7	0.08913	2.43943	0.08898	2.43519	0.08869	2.42109
	Low Priority Flows					
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.06415	0.99892	0.06363	0.99715	0.06189	0.98494
Demand 0.4 Utility 0.7	0.06415	0.99892	0.06428	0.99944	0.06265	0.98893
Demand 0.65 Utility 0.5	0.30482	2.10247	0.30228	2.08881	0.29568	2.05831
Demand 0.65 Utility 0.7	0.30482	2.10247	0.30339	2.09429	0.2953	2.0612
Ratio 4:1						
Demand 0.4 Utility 0.5	0.08939	0.83469	0.08887	0.83003	0.0874	0.8155
Demand 0.4 Utility 0.7	0.08939	0.83469	0.08939	0.83469	0.08825	0.82209
Demand 0.65 Utility 0.5	0.40111	1.58206	0.39685	1.55302	0.38263	1.51852
Demand 0.65 Utility 0.7	0.40111	1.58206	0.39741	1.55527	0.38388	1.5159
	Low Priority Flows					
With NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.05977	0.97553	0.05989	0.97703	0.05817	0.96586
Demand 0.4 Utility 0.7	0.05977	0.97553	0.06029	0.9769	0.05893	0.966
Demand 0.65 Utility 0.5	0.28563	2.06064	0.28721	2.07289	0.27932	2.02858
Demand 0.65 Utility 0.7	0.28563	2.06064	0.28911	2.08164	0.27864	2.03578
Ratio 4:1						
Demand 0.4 Utility 0.5	0.08142	0.81095	0.08132	0.80751	0.08027	0.79056
Demand 0.4 Utility 0.7	0.08142	0.81095	0.08186	0.81279	0.08054	0.79308
Demand 0.65 Utility 0.5	0.3715	1.52149	0.37302	1.52367	0.35592	1.47583
Demand 0.65 Utility 0.7	0.3715	1.52149	0.37126	1.51362	0.35796	1.47857

Table 8: Topology 4 Results

High Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.002	0.88783	0.00181	0.83843	0.00201	0.8697
Demand 0.4 Utility 0.7	0.002	0.88783	0.00181	0.8502	0.00197	0.87246
Demand 0.65 Utility 0.5	0.02375	2.21099	0.02297	2.13985	0.02337	2.15936
Demand 0.65 Utility 0.7	0.02375	2.21099	0.02312	2.1518	0.02321	2.15743
Ratio 4:1						
Demand 0.4 Utility 0.5	0.013	0.84408	0.01268	0.82097	0.01304	0.8365
Demand 0.4 Utility 0.7	0.013	0.84408	0.01277	0.824	0.01294	0.84105
Demand 0.65 Utility 0.5	0.09877	2.24878	0.09737	2.22705	0.09794	2.22941
Demand 0.65 Utility 0.7	0.09877	2.24878	0.09776	2.23079	0.09807	2.23414
High Priority Flows						
with NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.00176	0.7926	0.0016	0.75769	0.00177	0.79492
Demand 0.4 Utility 0.7	0.00176	0.7926	0.00168	0.76961	0.00173	0.79483
Demand 0.65 Utility 0.5	0.01971	1.99087	0.01926	1.93599	0.01936	1.95613
Demand 0.65 Utility 0.7	0.01971	1.99087	0.01924	1.94277	0.01951	1.96083
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01073	0.74698	0.01062	0.73525	0.01068	0.73468
Demand 0.4 Utility 0.7	0.01073	0.74698	0.01067	0.73623	0.01062	0.73716
Demand 0.65 Utility 0.5	0.08546	2.01342	0.08457	1.99414	0.08478	1.99551
Demand 0.65 Utility 0.7	0.08546	2.01342	0.08484	1.99991	0.08488	1.99735
Low Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.06465	0.93834	0.05678	0.86495	0.06619	0.9267
Demand 0.4 Utility 0.7	0.06465	0.93834	0.05785	0.88017	0.06528	0.92935
Demand 0.65 Utility 0.5	0.30824	1.93903	0.28302	1.83868	0.30474	1.89992
Demand 0.65 Utility 0.7	0.30824	1.93903	0.28429	1.85584	0.3064	1.9159
Ratio 4:1						
Demand 0.4 Utility 0.5	0.084	0.77186	0.07502	0.72645	0.08531	0.76479
Demand 0.4 Utility 0.7	0.084	0.77186	0.07528	0.73	0.0847	0.77497
Demand 0.65 Utility 0.5	0.40654	1.4305	0.3755	1.36443	0.39346	1.39583
Demand 0.65 Utility 0.7	0.40654	1.4305	0.37277	1.37265	0.39478	1.40391
Low Priority Flows						
With NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.05467	0.85986	0.04936	0.80609	0.05962	0.89278
Demand 0.4 Utility 0.7	0.05467	0.85986	0.05052	0.82418	0.05762	0.88024
Demand 0.65 Utility 0.5	0.27585	1.82663	0.25989	1.75143	0.28119	1.81836
Demand 0.65 Utility 0.7	0.27585	1.82663	0.25818	1.76528	0.28228	1.82772
Ratio 4:1						
Demand 0.4 Utility 0.5	0.06881	0.70047	0.06662	0.68095	0.06833	0.68339
Demand 0.4 Utility 0.7	0.06881	0.70047	0.06462	0.67687	0.06775	0.68421
Demand 0.65 Utility 0.5	0.36246	1.32747	0.34236	1.27955	0.34965	1.28933
Demand 0.65 Utility 0.7	0.36246	1.32747	0.34029	1.29119	0.34682	1.28811

Table 9: Topology 2 Secondary Results

High Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01312	0.56055	0.00133	0.28758	0.00127	0.28874
Demand 0.4 Utility 0.7	0.01312	0.56055	0.00131	0.29121	0.0013	0.29255
Demand 0.65 Utility 0.5	0.10493	1.59001	0.02925	1.02472	0.02984	1.03267
Demand 0.65 Utility 0.7	0.10493	1.59001	0.03003	1.05429	0.03018	1.06431
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01773	0.4995	0.00133	0.27696	0.00136	0.27748
Demand 0.4 Utility 0.7	0.01773	0.4995	0.00137	0.27964	0.00138	0.27981
Demand 0.65 Utility 0.5	0.11753	1.43245	0.03717	1.03639	0.03741	1.03839
Demand 0.65 Utility 0.7	0.11753	1.43245	0.03796	1.04944	0.03808	1.05055

High Priority Flows						
with NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01738	0.47939	0.00127	0.28073	0.00123	0.28206
Demand 0.4 Utility 0.7	0.01738	0.47939	0.0013	0.28454	0.00129	0.28603
Demand 0.65 Utility 0.5	0.11061	1.35616	0.0268	0.97483	0.02714	0.98237
Demand 0.65 Utility 0.7	0.11061	1.35616	0.02748	1.00682	0.02776	1.01425
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01247	0.54134	0.00124	0.27029	0.00125	0.27074
Demand 0.4 Utility 0.7	0.01247	0.54134	0.00126	0.27303	0.00128	0.27316
Demand 0.65 Utility 0.5	0.0992	1.52834	0.03459	0.98732	0.03479	0.98909
Demand 0.65 Utility 0.7	0.0992	1.52834	0.03536	0.99978	0.03544	1.00049

Low Priority Flows						
No NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01242	0.55496	0.00113	0.30371	0.00118	0.30483
Demand 0.4 Utility 0.7	0.01242	0.55496	0.00121	0.3075	0.0013	0.30856
Demand 0.65 Utility 0.5	0.10318	1.60288	0.03311	1.03075	0.03081	1.03199
Demand 0.65 Utility 0.7	0.10318	1.60288	0.03071	1.0579	0.03155	1.06664
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01644	0.4941	0.00127	0.27992	0.00125	0.28112
Demand 0.4 Utility 0.7	0.01644	0.4941	0.00128	0.28195	0.00138	0.28221
Demand 0.65 Utility 0.5	0.11821	1.42879	0.02925	0.95418	0.0301	0.95478
Demand 0.65 Utility 0.7	0.11821	1.42879	0.0297	0.97116	0.03137	0.97404

Low Priority Flows						
With NTO	No Update		Queue Update		Network Weatherman Update	
Ratio 1:1	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay	Dropped/Sent	Mean Delay
Demand 0.4 Utility 0.5	0.01638	0.47733	0.0011	0.29554	0.0011	0.2967
Demand 0.4 Utility 0.7	0.01638	0.47733	0.00121	0.30075	0.00129	0.30226
Demand 0.65 Utility 0.5	0.11003	1.36328	0.02911	0.97607	0.02795	0.97999
Demand 0.65 Utility 0.7	0.11003	1.36328	0.02801	1.00851	0.02871	1.01665
Ratio 4:1						
Demand 0.4 Utility 0.5	0.01141	0.53169	0.0011	0.2741	0.00106	0.27508
Demand 0.4 Utility 0.7	0.01141	0.53169	0.00103	0.27633	0.00115	0.27633
Demand 0.65 Utility 0.5	0.09729	1.52302	0.02657	0.90807	0.02696	0.9101
Demand 0.65 Utility 0.7	0.09729	1.52302	0.02682	0.92507	0.02868	0.92722

Bibliography

[1] T. Bonald and L. Massouli. 2001. “Impact of fairness on Internet performance,” SIGMETRICS Perform. Eval. Rev. 29, 1 (June 2001), 82-91.

[2] J. C.hen, M. Zhou, D. Li, and T. Sun. 2008. “A Priority Based Dynamic Adaptive Routing Protocol for Wireless Sensor Networks,” In Proceedings of the 2008 First International Conference on Intelligent Networks and Intelligent Systems (ICINIS '08). IEEE Computer Society, Washington, DC, USA, 160-164.

[3] K.C. Claffy, H.-W. Braun, G.C. Polyzos. "A Parameterizable Methodology for Internet traffic flow profiling," Selected Areas in Communications, IEEE Journal on , vol.13, no.8, pp.1481-1494, Oct 1995

[4] M. Compton. 2010. Improving the Quality of Service and Security of Military Networks with a Network Tasking Order Process. Electrical and Computer Engineering. WPAFB, Air Force Institute of Technology. Doctor of Philosophy.

[5] CRS Report for Congress. Network Centric Operations: Background and Oversight Issues for Congress., March 2007

[6] J.Domzal, A.Jajszczyk., "New Congestion Control Mechanisms for Flow-Aware Networks," Communications, 2008. ICC '08. IEEE International Conference on , vol., no., pp.12-16, 19-23 May 2008

[7] Elwalid, C. Jin, S. Low, I. Widjaja. "MATE: MPLS adaptive traffic engineering," INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.3, no., pp.1300-1309 vol.3, 2001

[8] E. Ekici, I.F. Akyildiz, M.D. Bender. "Datagram routing algorithm for LEO satellite networks," INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE , vol.2, no., pp.500-508 vol.2, 2000

[9] K. Fall. 2003. A delay-tolerant network architecture for challenged internets. In Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '03). ACM, New York, NY, USA, 27-34.

[10] L.K. Fleischer. "Approximating fractional multicommodity flow independent of the number of commodities," Foundations of Computer Science, 1999. 40th Annual Symposium on , vol., no., pp.24-31, 1999

[11] S. Ben Fred, T. Bonald, A. Proutiere, G. Régnié, and J. W. Roberts. 2001. Statistical bandwidth sharing: a study of congestion at flow level. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM '01). ACM, New York, NY, USA

[12] N. Garg, J. Konemann. "Faster and simpler algorithms for multicommodity flow and other fractional packing problems," Foundations of Computer Science, 1998. Proceedings.39th Annual Symposium on , vol., no., pp.300-309, 8-11 Nov 1998

[13] G. Karakostas. 2002. Faster approximation schemes for fractional multicommodity flow problems. In Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms (SODA '02). Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 166-173

[14] S. Keshav. 1991. A control-theoretic approach to flow control. SIGCOMMComput. Commun. Rev. 21, 4 (August 1991)

[15] Kortebi, S. Oueslati, J.W. Roberts. "Cross-protect: implicit service differentiation and admission control," High Performance Switching and Routing, 2004. HPSR. 2004 Workshop on , vol., no., pp. 56- 60, 2004

[16] J. Kurose, K. Ross. Computer Networking: A Top-Down Approach Addison-Wesley, 2010.

[17] J.L. Marzo, E. Calle, C. Scoglio, T. Anjah. "QoS online routing and MPLS multilevel protection: a survey," Communications Magazine, IEEE , vol.41, no.10, pp. 126- 132, Oct 2003

[18] S. McGrath, D. Chacòn, K. Whitebread. "Intelligent mobile agents in military command and control", in proceedings of the Workshop on Agents in Industry, in Autonomous Agents 2000 Conference, Barcelona, Spain, June 2000.

[19] D. Mitra, K.G. Ramakrishnan. "A case study of multiservice, multipriority traffic engineering design for data networks," Global Telecommunications Conference, 1999. GLOBECOM '99 , vol.1B, no., pp.1077-1083 vol. 1b, 1999

[20] S. Oueslati, J. Roberts. "A new direction for quality of service: flow-aware networking," Next Generation Internet Networks, 2005 , vol., no., pp. 226- 232, 18-20 April 2005

[21] M.K. Porwal, A. Yadav, S.V. Charhate. "Traffic Analysis of MPLS and Non MPLS Network including MPLS Signaling Protocols and Traffic Distribution in OSPF and MPLS," Emerging Trends in Engineering and Technology, 2008. ICETET '08. First International Conference on , vol., no., pp.187-192, 16-18 July 2008

[22] L. Massoulie and J. W. Roberts. "Bandwidth sharing and admission control for elastic traffic", Telecommun. Syst., vol. 15, p.185 , 2000.

[23] J. Roberts and S. Oueslati-Boulahia. "Quality of Service by Flow-Aware Networking", Phil Trans. R. Soc. Lond. A, pp.2197 - 2207 , 2000.

[24] F. Shahrokhi and D. W. Matula. 1990. The maximum concurrent flow problem. J. ACM 37, 2 (April 1990), 318-334. DOI=10.1145/77600.77620 http://doi.acm.org/10.1145/77600.77620

[25] N. Stuckey. 2007. Stochastic Estimation and Control of Queues within a Computer Network. Electrical and Computer Engineering. WPAFB, Air Force Institute of Technology. Master of Science.

[26] K. Thompson, G.J. Miller, R. Wilder. "Wide-area Internet traffic patterns and characteristics," Network, IEEE , vol.11, no.6, pp.10-23, Nov/Dec 1997

[27] X. Xiao, A. Hannan, B. Bailey, L.M. Ni. "Traffic engineering with MPLS in the Internet," Network, IEEE ,vol.14, no.2, pp.28-33, Mar/Apr 2000CRS Report for Congress. *Network Centric Operations: Background and Oversight Issues for Congress.* Order Code RL32411, Mar 2007

[28] E. W. Zegura, K. L. Calvert, S. Bhattacharjee."How to model an internetwork," INFOCOM '96. Fifteenth Annual Joint Conference of the IEEE Computer Societies. Networking the Next Generation. Proceedings IEEE , vol.2, no., pp.594-602 vol.2, 24-28 Mar 1996

Vita

Captain Timothy Jay Carbino is a student of the Air Force Institute of Technology pursuing a Masters Degree in Electrical Engineering. Captain Carbino graduated from the University of Florida with a Bachelor of Computer Engineering in 2006. He also obtained an Associate of Applied Science Degree from Fort Walton Community College in 2003.

REPORT DOCUMENTATION PAGE			*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
14 Jun 12	Master's Thesis	20 Aug 10 – 14 Jun 12

4. TITLE AND SUBTITLE
Adaptive Routing Algorithm for Priority Flows in a Network

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Carbino, Timothy J, Captain, USAF

5d. PROJECT NUMBER
12629P

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way
Wright-Patterson AFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GE/ENG/12-08

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Office of Scientific Research
Technical Advisor Dr. Robert Bonneau
875 N Randolph St, Ste 325 Rm 3112
Arlington, VA 22203
robert.bonneau@afosr.af.mil – (703) 426-9545

10. SPONSOR/MONITOR'S ACRONYM(S)
AFSOR/RSL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNILIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT
This research presents the development of an Adaptive Routing Algorithm for Priority (ARAP) flows in a Network. Devices in today's battle space require information to function. Additional bandwidth requirements for such devices place an increased burden on the already congested networks in the battle space. Some devices require real time information (high priority) and other devices will not require real time information (low priority). Existing protocols treat the network like an opaque entity and have little knowledge of user requirements. User requirement information is available in tactical networks and we can take advantage of the known requirements to better optimize network behavior. One such optimization is during times of congestion ARAP will enable better Quality of Service (QoS) for higher priority information. Mechanisms such as Network Tasking Order (NTO) and Network Weatherman (NWM) can provide this information to facilitate improved network behavior. The NTO gives advance knowledge of network state and NWM provides future estimates on utilization of specific network queues allowing for improved QoS guarantees.

15. SUBJECT TERMS
Network Weatherman, Network Tasking Order, Priority Routing, Quality of Service, Maximum Concurrent Mutli-Commodity Flow

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	UU	83	Kenneth Hopkinson, PHD, ENG
					19b. TELEPHONE NUMBER *(Include area code)* (937) 255-3636x4579;kenneth.hopkinson@afit.edu

Reset